

S01P0899US00

#4
J1017 U.S. PRO
09/875310
06/01

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office

出願年月日
Date of Application:

2000年 6月 8日

出願番号
Application Number:

特願2000-172681

出願人
Applicant(s):

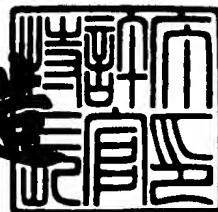
ソニー株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

2001年 5月 11日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造



出証番号 出証特2001-3040020

【書類名】 特許願
【整理番号】 0000342102
【提出日】 平成12年 6月 8日
【あて先】 特許庁長官 殿
【国際特許分類】 H03M 13/12

【発明者】

【住所又は居所】 東京都品川区北品川6丁目7番35号 ソニー株式会社内

【氏名】 宮内 俊之

【特許出願人】

【識別番号】 000002185

【氏名又は名称】 ソニー株式会社

【代表者】 出井 伸之

【代理人】

【識別番号】 100067736

【弁理士】

【氏名又は名称】 小池 晃

【選任した代理人】

【識別番号】 100086335

【弁理士】

【氏名又は名称】 田村 榮一

【選任した代理人】

【識別番号】 100096677

【弁理士】

【氏名又は名称】 伊賀 誠司

【手数料の表示】

【予納台帳番号】 019530

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9707387

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 復号装置及び復号方法

【特許請求の範囲】

【請求項1】 軟入力とされる受信値に基づいて任意のステートを通過する確率を対数表記した対数尤度を求め、上記対数尤度を用いて復号を行う復号装置であって、

上記対数尤度を与るために追加され、変数に対する1次元の関数で表される補正項を線形近似により算出する線形近似手段を備え、

上記線形近似手段は、少なくとも上記変数に乗算すべき上記関数の傾きを表す係数を2のべき乗を用いて表現して上記補正項を算出することを特徴とする復号装置。

【請求項2】 上記線形近似手段は、上記関数の傾きを表す係数を表現すべき数に応じて、入力されたデータの下位ビットから切り捨てることを特徴とする請求項1記載の復号装置。

【請求項3】 上記線形近似手段は、上記関数の傾きを表す係数を -2^{-k} で表現したとき、入力されたデータの下位1ビット目から下位kビット目までを切り捨てることを特徴とする請求項2記載の復号装置。

【請求項4】 上記線形近似手段は、上記関数の切片を表す係数を2のべき乗を用いて表現して上記補正項を算出することを特徴とする請求項1記載の復号装置。

【請求項5】 上記線形近似手段は、上記関数の切片を表す係数を $2^m - 1$ で表現して上記補正項を算出することを特徴とする請求項4記載の復号装置。

【請求項6】 上記線形近似手段は、上記関数の傾きを表す係数を -2^{-k} で表現したとき、入力されたnビットのデータの下位1ビット目から下位kビット目までを切り捨て、下位k+1ビット目から下位m+kビット目までのmビットを反転することを特徴とする請求項5記載の復号装置。

【請求項7】 上記補正項は、正値であること
を特徴とする請求項1記載の復号装置。

【請求項8】 上記線形近似手段は、上記補正項が負値として算出された場合
には、上記補正項を0とすること
を特徴とする請求項7記載の復号装置。

【請求項9】 上記対数尤度は、上記確率を自然対数を用いて対数表記したもの
であること
を特徴とする請求項1記載の復号装置。

【請求項10】 上記受信値毎に、符号の出力パターンと上記受信値により決
定される第1の確率を対数表記した第1の対数尤度を算出する第1の確率算出手
段と、
上記第1の対数尤度に基づいて、上記受信値毎に、符号化開始ステートから時
系列順に各ステートに至る第2の確率を対数表記した第2の対数尤度を算出する
第2の確率算出手段と、

上記第1の対数尤度に基づいて、上記受信値毎に、打ち切りステートから時系
列の逆順に各ステートに至る第3の確率を対数表記した第3の対数尤度を算出する
第3の確率算出手段とを備え、
上記第2の確率算出手段及び上記第3の確率算出手段は、それぞれ、上記線形
近似手段を有すること
を特徴とする請求項1記載の復号装置。

【請求項11】 上記第1の対数尤度と、上記第2の対数尤度と、上記第3の
対数尤度とを用いて、各時刻における軟出力を対数表記した対数軟出力を算出する
軟出力算出手段を備えること
を特徴とする請求項10記載の復号装置。

【請求項12】 上記対数軟出力は、上記軟出力を自然対数を用いて対数表記
したものであること
を特徴とする請求項11記載の復号装置。

【請求項13】 上記対数尤度は、上記確率の積演算を対数の和演算に置き換
えた計算を行うとともに、上記確率の和演算を対数の最大値演算と上記関数の演

算とを行うことで求められること

を特徴とする請求項1記載の復号装置。

【請求項14】 Log-B C J Rアルゴリズムに基づく最大事後確率復号を行うこと

を特徴とする請求項13記載の復号装置。

【請求項15】 叠み込み符号の復号を行うこと

を特徴とする請求項1記載の復号装置。

【請求項16】 軟入力とされる受信値に基づいて任意のステートを通過する確率を対数表記した対数尤度を求め、上記対数尤度を用いて復号を行う復号方法であって、

上記対数尤度を与るために追加され、変数に対する1次元の関数で表される補正項を線形近似により算出する線形近似工程を備え、

上記線形近似工程では、少なくとも上記変数に乘算すべき上記関数の傾きを表す係数を2のべき乗を用いて表現して上記補正項を算出すること

を特徴とする復号方法。

【請求項17】 上記線形近似工程では、上記関数の傾きを表す係数を表現するべき数に応じて、入力されたデータの下位ビットから切り捨てること

を特徴とする請求項16記載の復号方法。

【請求項18】 上記線形近似工程では、上記関数の傾きを表す係数を -2^{-k} で表現したとき、入力されたデータの下位1ビット目から下位kビット目までを切り捨てること

を特徴とする請求項17記載の復号方法。

【請求項19】 上記線形近似工程では、上記関数の切片を表す係数を2のべき乗を用いて表現して上記補正項を算出すること

を特徴とする請求項16記載の復号方法。

【請求項20】 上記線形近似工程では、上記関数の切片を表す係数を $2^m - 1$ で表現して上記補正項を算出すること

を特徴とする請求項19記載の復号方法。

【請求項21】 上記線形近似工程では、上記関数の傾きを表す係数を -2^{-k}

で表現したとき、入力された n ビットのデータの下位 1 ビット目から下位 k ビット目までを切り捨て、下位 $k + 1$ ビット目から下位 $m + k$ ビット目までの m ビットを反転すること

を特徴とする請求項 20 記載の復号方法。

【請求項 22】 上記補正項は、正值であること

を特徴とする請求項 16 記載の復号方法。

【請求項 23】 上記線形近似工程では、上記補正項が負値として算出された場合には、上記補正項を 0 とすること

を特徴とする請求項 22 記載の復号方法。

【請求項 24】 上記対数尤度は、上記確率を自然対数を用いて対数表記したものであること

を特徴とする請求項 16 記載の復号方法。

【請求項 25】 上記受信値毎に、符号の出力パターンと上記受信値により決定される第 1 の確率を対数表記した第 1 の対数尤度を算出する第 1 の確率算出工程と、

上記第 1 の対数尤度に基づいて、上記受信値毎に、符号化開始ステートから時系列順に各ステートに至る第 2 の確率を対数表記した第 2 の対数尤度を算出する第 2 の確率算出工程と、

上記第 1 の対数尤度に基づいて、上記受信値毎に、打ち切りステートから時系列の逆順に各ステートに至る第 3 の確率を対数表記した第 3 の対数尤度を算出する第 3 の確率算出工程とを備え、

上記第 2 の確率算出工程及び上記第 3 の確率算出工程は、それぞれ、上記線形近似工程を有すること

を特徴とする請求項 16 記載の復号方法。

【請求項 26】 上記第 1 の対数尤度と、上記第 2 の対数尤度と、上記第 3 の対数尤度とを用いて、各時刻における軟出力を対数表記した対数軟出力を算出する軟出力算出工程を備えること

を特徴とする請求項 25 記載の復号方法。

【請求項 27】 上記対数軟出力は、上記軟出力を自然対数を用いて対数表記

したものであること

を特徴とする請求項26記載の復号方法。

【請求項28】 上記対数尤度を、上記確率の積演算を対数の和演算に置き換えた計算を行うとともに、上記確率の和演算を対数の最大値演算と上記関数の演算とを行うことで求めること

を特徴とする請求項16記載の復号方法。

【請求項29】 Log-B C J Rアルゴリズムに基づく最大事後確率復号を行うこと

を特徴とする請求項28記載の復号方法。

【請求項30】 疊み込み符号の復号を行うこと

を特徴とする請求項16記載の復号方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、軟出力復号を行う復号装置及び復号方法に関する。

【0002】

【従来の技術】

近年、連接符号における内符号の復号出力や繰り返し復号法における各繰り返し復号動作の出力を軟出力とすることで、シンボル誤り率を小さくする研究がなされており、それに適した復号法に関する研究が盛んに行われている。例えば疊み込み符号等の所定の符号を復号した際のシンボル誤り率を最小にする方法としては、「Bahl, Cocke, Jelinek and Raviv, "Optimal decoding of linear codes for minimizing symbol error rate", IEEE Trans. Inf. Theory, vol. IT-20, pp. 284-287, Mar. 1974」に記載されているB C J Rアルゴリズムが知られている。このB C J Rアルゴリズムにおいては、復号結果として各シンボルを出力するのではなく、各シンボルの尤度を出力する。このような出力は、軟出力(soft-output)と呼ばれる。以下、このB C J Rアルゴリズムの内容について説明する。なお、以下の説明では、図15に示すように、デジタル情報を図示しない送信装置が備える符号化装置201により疊み込み符号化し、その出力を雜

音のある無記憶通信路202を介して図示しない受信装置に入力して、この受信装置が備える復号装置203により復号し、観測する場合を考える。

【0003】

まず、符号化装置201が備えるシフトレジスタの内容を表すM個のステート(遷移状態)を m (0, 1, ..., M-1)で表し、時刻tのステートを S_t で表す。また、1タイムスロットにkビットの情報が入力されるものとすると、時刻tにおける入力を $i_t = (i_{t1}, i_{t2}, \dots, i_{tk})$ で表し、入力系統を $I_1^T = (i_1, i_2, \dots, i_T)$ で表す。このとき、ステート m' からステートmへの遷移がある場合には、その遷移に対応する情報ビットを $i(m', m) = (i_1(m', m), i_2(m', m), \dots, i_k(m', m))$ で表す。さらに、1タイムスロットにnビットの符号が出力されるものとすると、時刻tにおける出力を $x_t = (x_{t1}, x_{t2}, \dots, x_{tn})$ で表し、出力系統を $X_1^T = (x_1, x_2, \dots, x_T)$ で表す。このとき、ステート m' からステートmへの遷移がある場合には、その遷移に対応する符号ビットを $x(m', m) = (x_1(m', m), x_2(m', m), \dots, x_n(m', m))$ で表す。

【0004】

符号化装置201による畠み込み符号化は、ステート $S_0 = 0$ から始まり、 X_1^T を出力して $S_T = 0$ で終了するものとする。ここで、各ステート間の遷移確率 $P_t(m | m')$ を次式(1)により定義する。

【0005】

【数1】

$$P_t(m | m') = Pr\{S_t = m | S_{t-1} = m'\} \quad (1)$$

【0006】

なお、上式(1)における右辺に示す $Pr\{A | B\}$ は、Bが生じた条件の下でのAが生じる条件付き確率である。この遷移確率 $P_t(m | m')$ は、次式(

2) に示すように、入力 i でステート m' からステート m へと遷移するときに、時刻 t での入力 i_t が i である確率 $P r \{ i_t = i \}$ と等しいものである。

【0007】

【数2】

$$P_r(m|m') = P r \{ i_t = i \} \quad (2)$$

【0008】

雑音のある無記憶通信路 202 は、 X_1^T を入力とし、 Y_1^T を出力する。ここで、1 タイムスロットに n ビットの受信値が出力されるものとすると、時刻 t における出力を $y_t = (y_{t1}, y_{t2}, \dots, y_{tn})$ で表し、 $Y_1^T = (y_1, y_2, \dots, y_T)$ で表す。雑音のある無記憶通信路 202 の遷移確率は、全ての t ($1 \leq t \leq T$) について、次式 (3) に示すように、各シンボルの遷移確率 $P r \{ y_j | x_j \}$ を用いて定義することができる。

【0009】

【数3】

$$P r \{ Y_1^T | X_1^T \} = \prod_{j=1}^T P r \{ y_j | x_j \} \quad (3)$$

【0010】

ここで、次式 (4) のように λ_{tj} を定義する。この次式 (4) に示す λ_{tj} は、 Y_1^T を受信した際の時刻 t での入力情報の尤度を表し、本来求めるべき軟出力である。

【0011】

【数4】

$$\lambda_{ij} = \frac{Pr\{i_{ij} = 1 | Y_1^T\}}{Pr\{i_{ij} = 0 | Y_1^T\}} \quad (4)$$

【0012】

BCJRアルゴリズムにおいては、次式(5)乃至次式(7)に示すような確率 α_t , β_t 及び γ_t を定義する。なお、 $Pr\{A; B\}$ は、AとBとがともに生じる確率を表すものとする。

【0013】

【数5】

$$\alpha_t(m) = Pr\{S_t = m; Y_1^t\} \quad (5)$$

【0014】

【数6】

$$\beta_t(m) = Pr\{Y_{t+1}^T | S_t = m\} \quad (6)$$

【0015】

【数7】

$$\gamma_t(m', m) = \Pr \left\{ S_t = m ; y_t | S_{t-1} = m' \right\} \quad (7)$$

【0016】

ここで、これらの確率 α_t , β_t 及び γ_t の内容について、符号化装置 201 における状態遷移図であるトレリスを図 16 を用いて説明する。同図において、 α_{t-1} は、符号化開始ステート $S_0 = 0$ から受信値をもとに時系列順に算出した時刻 $t - 1$ における各ステートの通過確率に対応する。また、 β_t は、符号化終了ステート $S_T = 0$ から受信値をもとに時系列の逆順に算出した時刻 t における各ステートの通過確率に対応する。さらに、 γ_t は、時刻 t における受信値と入力確率とをもとに算出した時刻 t にステート間を遷移する各枝の出力の受信確率に対応する。

【0017】

これらの確率 α_t , β_t 及び γ_t を用いると、軟出力 λ_{tj} は、次式 (8) のように表すことができる。

【0018】

【数8】

$$\lambda_{tj} = \frac{\sum_{\substack{m', m \\ i(m', m) = 1}} \alpha_i(m') \gamma_i(m', m) \beta_j(m)}{\sum_{\substack{m', m \\ i(m', m) = 0}} \alpha_i(m') \gamma_i(m', m) \beta_j(m)} \quad (8)$$

【0019】

ところで、 $t = 1, 2, \dots, T$ について、次式 (9) が成立する。

【0020】

【数9】

$$\alpha_t(m) = \sum_{m'=0}^{M-1} \alpha_{t-1}(m') \gamma_t(m', m)$$

ただし、 $\alpha_0(0) = 1, \alpha_0(m) = 0 (m \neq 0)$

(9)

【0021】

同様に、 $t = 1, 2, \dots, T$ について、次式（10）が成立する。

【0022】

【数10】

$$\beta_t(m) = \sum_{m'=0}^{M-1} \beta_{t+1}(m') \gamma_{t+1}(m, m')$$

ただし、 $\beta_T(0) = 1, \beta_T(m) = 0 (m \neq 0)$

(10)

【0023】

さらに、 γ_t について、次式（11）が成立する。

【0024】

【数11】

$$\gamma_t(m', m) = \begin{cases} P_i(m | m') \cdot Pr\{y_t | x(m', m)\} \\ = Pr\{i_t = i(m', m)\} \cdot Pr\{y_t | x(m', m)\} & (11) \\ & : \text{入力 } i \text{ で } m' \text{ から } m \text{ へ遷移する場合} \\ 0 & : \text{入力 } i \text{ で } m' \text{ から } m \text{ へ遷移しない場合} \end{cases}$$

【0025】

したがって、復号装置203は、BCJRアルゴリズムを適用して軟出力復号を行う場合には、これらの関係に基づいて、図17に示す一連の工程を経ることにより軟出力 λ_t を求める。

【0026】

まず、復号装置203は、同図に示すように、ステップS201において、 y_t を受信する毎に、上式(9)及び上式(11)を用いて、確率 $\alpha_t(m)$ 及び $\gamma_t(m', m)$ を算出する。

【0027】

続いて、復号装置203は、ステップS202において、系列 Y_1^T の全てを受信した後に、上式(10)を用いて、全ての時刻 t における各ステート m について、確率 $\beta_t(m)$ を算出する。

【0028】

そして、復号装置203は、ステップS203において、ステップS201及びステップS202において算出した確率 α_t 、 β_t 及び γ_t を上式(8)に代入し、各時刻 t における軟出力 λ_t を算出する。

【0029】

復号装置203は、このような一連の処理を経ることによって、BCJRアルゴリズムを適用した軟出力復号を行うことができる。

【0030】

ところで、このようなBCJRアルゴリズムにおいては、確率を直接値として保持して演算を行う必要があり、積演算を含むために演算量が大きいという問題があった。そこで、演算量を削減する手法として、「Robertson, Villebrun and Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the domain" , IEEE Int. Conf. on Communications, pp. 1009-1013, June 1995」に記載されているMax-Log-MAPアルゴリズム及びLog-MAPアルゴリズム（以下、Max-Log-BCJRアルゴリズム及びLog-BCJRアルゴリズムと称する。）がある。

【0031】

まず、Max-Log-BCJRアルゴリズムについて説明する。Max-Log-BCJRアルゴリズムは、確率 α_t , β_t 並びに γ_t 、及び軟出力 λ_t を自然対数を用いて対数表記し、次式(12)に示すように、確率の積演算を対数の和演算に置き換えるとともに、次式(13)に示すように、確率の和演算を対数の最大値演算で近似するものである。なお、次式(13)に示すmax(x, y)は、x, yのうち大きい値を有するものを選択する関数である。

【0032】

【数12】

$$\log(e^x \cdot e^y) = x + y \quad (12)$$

【0033】

【数13】

$$\log(e^x + e^y) = \max(x, y) \quad (13)$$

【0034】

ここで、記載を簡略化するため、自然対数を I と略記し、 α_t , β_t , γ_t , λ_t の自然対数值を、それぞれ、次式(14)に示すように、 $I\alpha_t$, $I\beta_t$, $I\gamma_t$, $I\lambda_t$ と表すものとする。

【0035】

【数14】

$$\begin{cases} I\alpha_t(m) = \log(\alpha_t(m)) \\ I\beta_t(m) = \log(\beta_t(m)) \\ I\gamma_t(m) = \log(\gamma_t(m)) \\ I\lambda_t = \log \lambda_t \end{cases} \quad (14)$$

【0036】

Max-Llog-BCLRアルゴリズムにおいては、これらの対数尤度(log likelihood) $I\alpha_t$, $I\beta_t$, $I\gamma_t$ を、それぞれ、次式(15)乃至次式(17)に示すように近似する。ここで、次式(15)における右辺のステート m' に

おける最大値 m_{\max} は、ステート m への遷移が存在するステート m' の中で求めるものとし、次式(16)における右辺のステート m' における最大値 m_{\max} は、ステート m からの遷移が存在するステート m' の中で求めるものとする。

【0037】

【数15】

$$I\alpha_i(m) \simeq \max_{m'} \left(I\alpha_{i-1}(m') + I\gamma_i(m', m) \right) \quad (15)$$

【0038】

【数16】

$$I\beta_i(m) \simeq \max_{m'} \left(I\beta_{i+1}(m') + I\gamma_{i+1}(m, m') \right) \quad (16)$$

【0039】

【数17】

$$I\gamma_i(m', m) = \log \left(Pr \left\{ i_i = i(m', m) \right\} \right) + \log \left(Pr \left\{ y_i | x(m', m) \right\} \right) \quad (17)$$

【0040】

また、Max-Log-BCLRアルゴリズムにおいては、対数軟出力 $I\lambda_t$ についても同様に、次式(18)に示すように近似する。ここで、次式(18)における右辺第1項の最大値 m_{\max} は、入力が“1”的ときにステート m への遷

移が存在するステート m' の中で求め、第2項の最大値 \max は、入力が“0”のときにステート m への遷移が存在するステート m' の中で求めるものとする。

【0041】

【数18】

$$\begin{aligned} I\lambda_{tj} &= \max_{\substack{m', m \\ i(m', m)=1}} \left(I\alpha_{t-1}(m') + I\gamma_i(m', m) + I\beta_i(m) \right) \\ &\quad - \max_{\substack{m', m \\ i(m', m)=0}} \left(I\alpha_{t-1}(m') + I\gamma_i(m', m) + \beta_i(m) \right) \end{aligned} \quad (18)$$

【0042】

したがって、復号装置203は、Max-Log-BCLRアルゴリズムを適用して軟出力復号を行う場合には、これらの関係に基づいて、図18に示す一連の工程を経ることにより軟出力 λ_t を求める。

【0043】

まず、復号装置203は、同図に示すように、ステップS211において、 y_t を受信する毎に、上式(15)及び上式(17)を用いて、対数尤度 $I\alpha_t(m)$ 及び $I\gamma_t(m', m)$ を算出する。

【0044】

続いて、復号装置203は、ステップS212において、系列 Y_1^T の全てを受信した後に、上式(16)を用いて、全ての時刻 t における各ステート m について、対数尤度 $I\beta_t(m)$ を算出する。

【0045】

そして、復号装置203は、ステップS213において、ステップS211及びステップS212において算出した対数尤度 $I\alpha_t$, $I\beta_t$ 及び $I\gamma_t$ を上式(18)に代入し、各時刻 t における対数軟出力 $I\lambda_t$ を算出する。

【0046】

復号装置203は、このような一連の処理を経ることによって、Max-Log-BCLRアルゴリズムを適用した軟出力復号を行うことができる。

【0047】

このように、Max-Log-BCHアルゴリズムは、積演算が含まれないことから、BCHアルゴリズムと比較して、演算量を大幅に削減することができる。

【0048】

つぎに、Log-BCHアルゴリズムについて説明する。Log-BCHアルゴリズムは、Max-Log-BCHアルゴリズムによる近似の精度をより向上させたものである。具体的には、Log-BCHアルゴリズムは、上式(13)に示した確率の和演算を次式(19)に示すように補正項を追加することで変形し、和演算の正確な対数値を求めるものである。ここでは、このような補正をlog-sum補正と称するものとする。

【0049】

【数19】

$$\log(e^x + e^y) = \max(x, y) + \log\left(1 + e^{-|x-y|}\right) \quad (19)$$

【0050】

ここで、上式(19)における左辺に示す演算をlog-sum演算と称するものとし、このlog-sum演算の演算子を、「S. S. Pietrobon, "Implementation and performance of a turbo/MAP decoder", Int. J. Satellite Commu., vol. 16, pp. 23-46, Jan.-Feb. 1998」に記載されている記数法を踏襲し、次式(20)に示すように、便宜上“#”(ただし、同論文中では、“E”。)と表すものとする。さらに、log-sum演算の累積加算演算の演算子を、次式(21)に示すように、“#Σ”(ただし、同論文中では、“E”。)と表すものとする。

【0051】

【数20】

$$x \# y = \log(e^x + e^y) \quad (20)$$

【0052】

【数21】

$$\# \sum_{i=0}^{M-1} x_i = \left(\left(\cdots \left((x_0 \# x_1) \# x_2 \right) \cdots \right) \# x_{M-1} \right) \quad (21)$$

【0053】

これらの演算子を用いると、LOG-B C J R アルゴリズムにおける対数尤度 $I\alpha_t$, $I\beta_t$ 及び対数軟出力 $I\lambda_t$ は、それぞれ、次式 (22) 乃至次式 (24) に示すように表すことができる。なお、対数尤度 $I\gamma_t$ は、上式 (17) で表されるため、ここでは、その記述を省略する。

【0054】

【数22】

$$I\alpha_t(m) = \# \sum_{m'=0}^{M-1} \left(I\alpha_{t-1}(m') + I\gamma_t(m', m) \right) \quad (22)$$

【0055】

【数23】

$$I\beta_t(m) = \# \sum_{m'=0}^{M-1} \left(I\beta_{t+1}(m') + I\gamma_{t+1}(m, m') \right) \quad (23)$$

【0056】

【数24】

$$\begin{aligned} I\lambda_{tj}(m) &= \# \sum_{\substack{m', m \\ i_j(m', m) = 1}} \left(I\alpha_{t-1}(m') + I\gamma_t(m', m) + I\beta_t(m) \right) \\ &\quad - \# \sum_{\substack{m', m \\ i_j(m', m) = 0}} \left(I\alpha_{t-1}(m') + I\gamma_t(m', m) + I\beta_t(m) \right) \end{aligned} \quad (24)$$

【0057】

なお、上式(22)における右辺のステート m' における $\log - \text{sum}$ 演算の累積加算演算は、ステート m への遷移が存在するステート m' の中で求めるものとし、上式(23)における右辺のステート m' における $\log - \text{sum}$ 演算の累積加算演算は、ステート m からの遷移が存在するステート m' の中で求めるものとする。また、上式(24)における右辺第1項の $\log - \text{sum}$ 演算の累積加算演算は、入力が “1” のときにステート m への遷移が存在するステート m' の中で求め、第2項の $\log - \text{sum}$ 演算の累積加算演算は、入力が “0” のときにステート m への遷移が存在するステート m' の中で求めるものとする。

【0058】

したがって、復号装置203は、 $\text{Log}-\text{BCJR}$ アルゴリズムを適用して軟出力復号を行う場合には、これらの関係に基づいて、先に図18に示した一連の工程を経ることにより軟出力 λ_t を求めることができる。

【0059】

まず、復号装置203は、同図に示すように、ステップS211において、 y

t を受信する毎に、上式(22)及び上式(17)を用いて、対数尤度 $I \alpha_t$ (m) 及び $I \gamma_t$ (m' , m) を算出する。

【0060】

続いて、復号装置203は、ステップS212において、系列 Y_1^T の全てを受信した後に、上式(23)を用いて、全ての時刻 t における各ステート m について、対数尤度 $I \beta_t$ (m) を算出する。

【0061】

そして、復号装置203は、ステップS213において、ステップS211及びステップS212において算出した対数尤度 $I \alpha_t$, $I \beta_t$ 及び $I \gamma_t$ を上式(24)に代入し、各時刻 t における対数軟出力 $I \lambda_t$ を算出する。

【0062】

復号装置203は、このような一連の処理を経ることによって、L_og-B_CJ_Rアルゴリズムを適用した軟出力復号を行うことができる。なお、上式(19)において、右辺第2項に示す補正項は、変数 $|x - y|$ に対する1次元の関数で表されることから、復号装置203は、この値を図示しないROM(Read Only Memory)等にテーブルとして予め記憶させておくことによって、正確な確率計算を行うことができる。

【0063】

このようなL_og-B_CJ_Rアルゴリズムは、Max-L_og-B_CJ_Rアルゴリズムと比較すると演算量は増えるものの積演算を含むものではなく、その出力は、量子化誤差を除けば、B_CJ_Rアルゴリズムの軟出力の対数値そのものに他ならない。

【0064】

【発明が解決しようとする課題】

ところで、上述したlog-sum補正の方法としては、上述したように、補正項の値をテーブル化しておく方法の他に、変数 $|x - y|$ との関係をいわゆる2次近似により近似する2次近似法、変数 $|x - y|$ を任意の区間に分割して各区間毎に所定の値を与える区間分割法等がある。補正項の値をテーブル化しておく方法を含め、これらの方は、補正項の値をいかに正確に求めるかといった性

能を重視した補正である。しかしながら、これら的方法は、回路規模の増大や速度の遅延を招くといった問題があった。

【0065】

そこで、 $\log - \text{sum}$ 補正の方法としては、速度を重視した方法が検討されている。この方法としては、変数 $|x - y|$ との関係をいわゆる線形近似により近似する線形近似法、変数 $|x - y|$ における所定の区間の値を所定の閾値で決定する閾値近似法がある。

【0066】

線形近似法は、図19(A)に示すように、曲線Cに示す関数 $F = \log \{1 + e^{-|x-y|}\}$ を直線Lに示す線形関数で近似するものである。同図においては、直線Lとして、 $F = -0.3(|x-y|) + \log 2$ を用いており、この場合、約0.1dB程度の劣化で補正項を求めることができる。

【0067】

また、閾値近似法は、同図(B)に示すように、曲線Cに示す関数 $F = \log \{1 + e^{-|x-y|}\}$ を曲線Tに示す階段関数で近似するものである。同図においては、曲線Tとして、 $0 \leq |x-y| < 1$ の区間では $\log 2$ を与え、 $|x-y| \geq 1$ の区間では0を与えるような関数を用いている。この場合、約0.2dB程度の劣化で補正項を求めることができる。

【0068】

このように、 $\log - \text{sum}$ 補正としては、様々な方法が検討されているが、一方で、未だ改善の余地が残るのが現状である。

【0069】

本発明は、このような実情に鑑みてなされたものであり、速度を重視した線形近似により $\log - \text{sum}$ 補正を行い、性能を劣化させることなく、高速化を図ることができる復号装置及び復号方法を提供することを目的とする。

【0070】

【課題を解決するための手段】

上述した目的を達成する本発明にかかる復号装置は、軟入力とされる受信値に基づいて任意のステートを通過する確率を対数表記した対数尤度を求め、この対

数尤度を用いて復号を行う復号装置であって、対数尤度を与えるために追加され、変数に対する1次元の関数で表される補正項を線形近似により算出する線形近似手段を備え、この線形近似手段は、少なくとも変数に乘算すべき関数の傾きを表す係数を2のべき乗を用いて表現して補正項を算出することを特徴としている。

【0071】

このような本発明にかかる復号装置は、線形近似手段によって、2のべき乗を用いて表現された係数を有する関数で表される補正項を算出する。

【0072】

また、上述した目的を達成する本発明にかかる復号方法は、軟入力とされる受信値に基づいて任意のステートを通過する確率を対数表記した対数尤度を求め、この対数尤度を用いて復号を行う復号方法であって、対数尤度を与えるために追加され、変数に対する1次元の関数で表される補正項を線形近似により算出する線形近似工程を備え、この線形近似工程では、少なくとも変数に乘算すべき関数の傾きを表す係数を2のべき乗を用いて表現して補正項を算出することを特徴としている。

【0073】

このような本発明にかかる復号方法は、線形近似工程にて、2のべき乗を用いて表現された係数を有する関数で表される補正項を算出する。

【0074】

【発明の実施の形態】

以下、本発明を適用した具体的な実施の形態について図面を参照しながら詳細に説明する。

【0075】

この実施の形態は、図1に示すように、デジタル情報を図示しない送信装置が備える符号化装置1により畳み込み符号化し、その出力を雑音のある無記憶通信路2を介して図示しない受信装置に入力して、この受信装置が備える復号装置3により復号する通信モデルに適用したデータ送受信システムである。

【0076】

このデータ送受信システムにおいて、復号装置3は、符号化装置1により畳み込み符号化がなされた符号の復号を行うものであって、「Robertson, Villebrun and Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the domain", IEEE Int. Conf. on Communications, pp. 1009-1013, June 1995」に記載されているLog-MAPアルゴリズム（以下、Log-BCLRアルゴリズムと称する。）に基づく最大事後確率（Maximum A Posteriori probability；以下、MAPと記す。）復号を行うものとして構成され、いわゆる確率 α_t , β_t , γ_t 、及び軟出力（soft-output） λ_t を自然対数を用いて対数表記した対数尤度（log likelihood） $I\alpha_t$, $I\beta_t$, $I\gamma_t$ 、及び対数軟出力 $I\lambda_t$ を線形近似によるlog-sum補正を行うことで求めるものである。

【0077】

なお、以下では、符号化装置1が備えるシフトレジスタの内容を表すM個のステート（遷移状態）を m （0, 1, …, M-1）で表し、時刻tのステートを S_t で表す。また、1タイムスロットにkビットの情報が入力されるものとすると、時刻tにおける入力を $i_t = (i_{t1}, i_{t2}, \dots, i_{tk})$ で表し、入力系統を $I_1^T = (i_1, i_2, \dots, i_T)$ で表す。このとき、ステート m' からステートmへの遷移がある場合には、その遷移に対応する情報ビットを $i(m', m) = (i_1(m', m), i_2(m', m), \dots, i_k(m', m))$ で表す。さらに、1タイムスロットにnビットの符号が出力されるものとすると、時刻tにおける出力を $x_t = (x_{t1}, x_{t2}, \dots, x_{tn})$ で表し、出力系統を $X_1^T = (x_1, x_2, \dots, x_T)$ で表す。このとき、ステート m' からステートmへの遷移がある場合には、その遷移に対応する符号ビットを $x(m', m) = (x_1(m', m), x_2(m', m), \dots, x_n(m', m))$ で表す。さらにまた、無記憶通信路2は、 X_1^T を入力とし、 Y_1^T を出力するものとする。ここで、1タイムスロットにnビットの受信値が出力されるものとすると、時刻tにおける出力を $y_t = (y_{t1}, y_{t2}, \dots, y_{tn})$ で表し、 $Y_1^T = (y_1, y_2, \dots, y_T)$ で表す。

【0078】

符号化装置1は、例えば図2に示すように、3つの排他的論理和回路11，13，15と、2つのシフトレジスタ12，14とを有し、拘束長が“3”の畳み込み演算を行うものとして構成される。

【0079】

排他的論理和回路11は、1ビットの入力データ*i_{t1}*と、排他的論理和回路13から供給されるデータとを用いて排他的論理和演算を行い、演算結果をシフトレジスタ12及び排他的論理和回路15に供給する。

【0080】

シフトレジスタ12は、保持している1ビットのデータを排他的論理和回路13及びシフトレジスタ14に供給し続ける。そして、シフトレジスタ12は、クロックに同期させて、排他的論理和回路11から供給される1ビットのデータを新たに保持し、このデータを排他的論理和回路13及びシフトレジスタ14に新たに供給する。

【0081】

排他的論理和回路13は、シフトレジスタ12，14から供給されるデータを用いて排他的論理和演算を行い、演算結果を排他的論理和回路11に供給する。

【0082】

シフトレジスタ14は、保持している1ビットのデータを排他的論理和回路13，15に供給し続ける。そして、シフトレジスタ14は、クロックに同期させて、シフトレジスタ12から供給される1ビットのデータを新たに保持し、このデータを排他的論理和回路13，15に新たに供給する。

【0083】

排他的論理和回路15は、排他的論理和回路11から供給されるデータと、シフトレジスタ14から供給されるデータとを用いて排他的論理和演算を行い、演算結果を2ビットの出力データ*x_t*のうちの1ビットの出力データ*x_{t2}*として外部に出力する。

【0084】

このような符号化装置1は、1ビットの入力データ*i_{t1}*を入力すると、この入力データ*i_{t1}*を、2ビットの出力データ*x_t*のうちの組織成分の1ビットの出力

データ x_{t_1} として、そのまま外部に出力するとともに、入力データ i_{t_1} に対して再帰的畳み込み演算を行い、演算結果を 2 ビットの出力データ x_t のうちの他方の 1 ビットの出力データ x_{t_2} として外部に出力する。すなわち、符号化装置 1 は、符号化率が “1／2” の再帰的組織畳み込み演算を行い、出力データ x_t を外部に出力する。

【0085】

この符号化装置 1 におけるトレリスを記述すると、図 3 に示すようになる。同図において、破線で示すバスは、入力データ i_{t_1} が “0” の場合を示し、実線で示すバスは、入力データ i_{t_1} が “1” の場合を示している。また、各バスに付与されているラベルは、2 ビットの出力データ x_t を示している。ここでは、ステートは、シフトレジスタ 1 2 の内容とシフトレジスタ 1 4 の内容とを順次並べたものであり、“00”、“10”、“01”、“11”的ステート番号を、それぞれ、“0”、“1”、“2”、“3”と表している。このように、符号化装置 1 におけるステート数 M は 4 となり、トレリスは、各ステートから次時刻におけるステートへと 2 本のバスが到達する構造を有する。なお、以下の説明では、各ステート番号に対応するステートを指示する場合には、それぞれ、ステート 0、ステート 1、ステート 2、ステート 3 と称するものとする。

【0086】

このような符号化装置 1 により符号化された出力データ x_t は、無記憶通信路 2 を介して受信装置に出力される。

【0087】

一方、復号装置 3 は、図 4 に示すように、各部を制御するコントローラ 3 1 と、第 1 の対数尤度である対数尤度 I_γ を算出して記憶する第 1 の確率算出手段である I_γ 算出・記憶回路 3 2 と、第 2 の対数尤度である対数尤度 I_α を算出して記憶する第 2 の確率算出手段である I_α 算出・記憶回路 3 3 と、第 3 の対数尤度である対数尤度 I_β を算出して記憶する第 3 の確率算出手段である I_β 算出・記憶回路 3 4 と、対数軟出力 I_λ_t を算出する軟出力算出手段である軟出力算出回路 3 5 とを備える。この復号装置 3 は、無記憶通信路 2 上で発生したノイズの影響によりアナログ値をとり軟入力 (soft-input) とされる受信値 y_t から対数軟

出力 $I \lambda_t$ を求めることによって、符号化装置 1 における入力データ i_{t1} を推定するものである。

【0088】

コントローラ 31 は、 $I \gamma$ 算出・記憶回路 32、 $I \alpha$ 算出・記憶回路 33 及び $I \beta$ 算出・記憶回路 34 に対して、 それぞれ、 コントロール信号 $SC\gamma$ 、 $SC\alpha$ 及び $SC\beta$ を供給し、 各部の動作を制御する。

【0089】

$I \gamma$ 算出・記憶回路 32 は、 コントローラ 31 から供給されたコントロール信号 $SC\gamma$ による制御の下に、 受信値 y_t と、 事前確率情報 (*a priori probability information*) $P r_t$ とを用いて、 受信値 y_t 每に、 次式 (25) に示す演算を行い、 各時刻 t における対数尤度 $I \gamma_t$ を算出して記憶する。 すなわち、 $I \gamma$ 算出・記憶回路 32 は、 受信値 y_t 每に、 符号の出力パターンと受信値により決定される確率 γ を対数表記した対数尤度 $I \gamma$ を算出する。

【0090】

【数25】

$$I \gamma_t(m', m) = \log \left(P r \left\{ i_t = i(m', m) \right\} \right) + \log \left(P r \left\{ y_t | x(m', m) \right\} \right) \quad (25)$$

【0091】

なお、 事前確率情報 $P r_t$ は、 次式 (26) に示すように、 入力データ i_{t1} が “1” である確率 $P r \{ i_{t1} = 1 \}$ 又は入力データ i_{t1} が “0” である確率 $P r \{ i_{t1} = 0 \}$ として与えられる。 また、 事前確率情報 $P r_t$ は、 確率 $P r \{ i_{t1} = 1 \}$ と確率 $P r \{ i_{t1} = 0 \}$ の比の自然対数値である対数尤度比 (*log likelihood ratio*) を入力し、 確率 $P r \{ i_{t1} = 1 \}$ と確率 $P r \{ i_{t1} = 0 \}$ の和が “1” であることを考慮して、 確率 $P r \{ i_{t1} = 1 \}$ 又は確率 $P r \{ i_{t1} = 0 \}$ として求められてもよい。

【0092】

【数26】

$$Pr_i = \begin{cases} \log Pr\{i_n = 1\} \\ \log Pr\{i_n = 0\} \end{cases} \quad (26)$$

【0093】

そして、 $I\gamma$ 算出・記憶回路32は、記憶した対数尤度 $I\gamma_t$ を $I\alpha$ 算出・記憶回路33、 $I\beta$ 算出・記憶回路34及び軟出力算出回路35に供給する。このとき、 $I\gamma$ 算出・記憶回路32は、 $I\alpha$ 算出・記憶回路33、 $I\beta$ 算出・記憶回路34及び軟出力算出回路35のそれぞれにおける処理に適した順序で対数尤度 $I\gamma_t$ を供給する。なお、以下の説明では、 $I\gamma$ 算出・記憶回路32から $I\alpha$ 算出・記憶回路33に供給される対数尤度 $I\gamma_t$ を $I\gamma(\alpha)$ と表し、 $I\gamma$ 算出・記憶回路32から $I\beta$ 算出・記憶回路34に供給される対数尤度 $I\gamma_t$ を $I\gamma(\beta 1)$ 、 $I\gamma(\beta 2)$ と表し、 $I\gamma$ 算出・記憶回路32から軟出力算出回路35に供給される対数尤度 $I\gamma_t$ を $I\gamma(\lambda)$ と表すものとする。

【0094】

$I\alpha$ 算出・記憶回路33は、コントローラ31から供給されたコントロール信号 $SC\alpha$ による制御の下に、 $I\gamma$ 算出・記憶回路32から供給された対数尤度 $I\gamma(\alpha)$ を用いて、次式(27)に示す演算を行い、各時刻 t における対数尤度 $I\alpha_t$ を算出して記憶する。なお、次式(27)における演算子“#”は、いわゆる $log - sum$ 演算を示すものであり、入力“0”でステート m' からステート m へと遷移するときにおける対数尤度と、入力“1”でステート m' からステート m へと遷移するときにおける対数尤度との $log - sum$ 演算を示すものである。より具体的には、 $I\alpha$ 算出・記憶回路33は、次式(28)に示す演算を行うことによって、各時刻 t における対数尤度 $I\alpha_t$ を算出する。すなわち、 $I\alpha$ 算出・記憶回路33は、対数尤度 $I\gamma$ に基づいて、受信値 y_t 毎に、符号化開始ステートから時系列順に各ステートに至る確率 α を対数表記した対数尤度

$I\alpha$ を算出する。そして、 $I\alpha$ 算出・記憶回路33は、記憶した対数尤度 $I\alpha_t$ を軟出力算出回路35に供給する。このとき、 $I\alpha$ 算出・記憶回路33は、軟出力算出回路35における処理に適した順序で対数尤度 $I\alpha_t$ を供給する。なお、以下の説明では、 $I\alpha$ 算出・記憶回路33から軟出力算出回路35に供給される対数尤度 $I\alpha_t$ を $I\alpha(\lambda)$ と表すものとする。

【0095】

【数27】

$$I\alpha_t(m) = \begin{cases} I\alpha_{t-1}(m') + I\gamma_t(m', m) \\ \#(I\alpha_{t-1}(m'') + I\gamma_t(m'', m)) \end{cases} \quad (27)$$

【0096】

【数28】

$$I\alpha_t(m) = \max \left(I\alpha_{t-1}(m') + I\gamma_t(m', m), I\alpha_{t-1}(m'') + I\gamma_t(m'', m) \right) \\ + \log \left(\frac{e^{-\left| (I\alpha_{t-1}(m') + I\gamma_t(m', m)) - (I\alpha_{t-1}(m'') + I\gamma_t(m'', m)) \right|}}{1 + e^{-\left| (I\alpha_{t-1}(m') + I\gamma_t(m', m)) - (I\alpha_{t-1}(m'') + I\gamma_t(m'', m)) \right|}} \right) \quad (28)$$

【0097】

$I\beta$ 算出・記憶回路34は、コントローラ31から供給されたコントロール信号 $SC\beta$ による制御の下に、 $I\gamma$ 算出・記憶回路32から供給された対数尤度 $I\gamma(\beta 1)$, $I\gamma(\beta 2)$ を用いて、次式(29)に示す演算を行い、各時刻における2系統の対数尤度 $I\beta_t$ を並列的に算出して記憶する。なお、次式(29)における演算子“#”は、上述したように、log-sum演算を示すものであり、入力“0”でステート m' からステート m へと遷移するときにおける対数尤度と、入力“1”でステート m'' からステート m へと遷移するときにおける対数尤度とのlog-sum演算を示すものである。より具体的には、 $I\beta$ 算出

・記憶回路34は、次式(30)に示す演算を行うことによって、各時刻 t における対数尤度 $I\beta_t$ を算出する。すなわち、 $I\beta$ 算出・記憶回路34は、対数尤度 $I\gamma$ に基づいて、受信値 y_t 毎に、打ち切りステートから時系列の逆順に各ステートに至る確率 β を対数表記した対数尤度 $I\beta$ を算出する。そして、 $I\beta$ 算出・記憶回路34は、記憶した対数尤度 $I\beta_t$ のうち、1系統の対数尤度 $I\beta_t$ を軟出力算出回路35に供給する。このとき、 $I\beta$ 算出・記憶回路34は、軟出力算出回路35における処理に適した順序で対数尤度 $I\beta_t$ を供給する。なお、以下の説明では、 $I\beta$ 算出・記憶回路34から軟出力算出回路35に供給される対数尤度 $I\beta_t$ を $I\beta(\lambda)$ と表すものとする。

【0098】

【数29】

$$I\beta_t(m) = \begin{cases} I\beta_{t+1}(m') + I\gamma_{t+1}(m, m') \\ \# \left(I\beta_{t+1}(m'') + I\gamma_{t+1}(m, m'') \right) \end{cases} \quad (29)$$

【0099】

【数30】

$$I\beta_t(m) = \max \left(I\beta_{t+1}(m') + I\gamma_{t+1}(m, m'), I\beta_{t+1}(m'') + I\gamma_{t+1}(m, m'') \right) \\ + \log \left(\frac{e^{-\left| (I\beta_{t+1}(m') + I\gamma_{t+1}(m, m')) - (I\beta_{t+1}(m'') + I\gamma_{t+1}(m, m'')) \right|}}{1 + e^{-\left| (I\beta_{t+1}(m') + I\gamma_{t+1}(m, m')) - (I\beta_{t+1}(m'') + I\gamma_{t+1}(m, m'')) \right|}} \right) \quad (30)$$

【0100】

軟出力算出回路35は、 $I\gamma$ 算出・記憶回路32から供給された対数尤度 $I\gamma(\lambda)$ と、 $I\alpha$ 算出・記憶回路33から供給された対数尤度 $I\alpha(\lambda)$ と、 $I\beta$ 算出・記憶回路34から供給された対数尤度 $I\beta(\lambda)$ とを用いて、次式(31)に示す演算を行い、各時刻における対数軟出力 $I\lambda_t$ を算出して記憶する。そ

して、軟出力算出回路3 5は、記憶した対数軟出力 $I \lambda_t$ を時系列順に並べ替えた後、外部に出力する。なお、次式(31)における演算子“#Σ”は、上述した演算子“#”で表される $\text{log} - \text{sum}$ 演算の累積加算演算を示すものである。

【0101】

【数31】

$$\begin{aligned} I \lambda_t = & \# \sum_{\substack{m', m \\ i(m', m) = 1}} \left(I \alpha_{t-1}(m') + I \gamma_t(m', m) + I \beta_t(m) \right) \\ & - \# \sum_{\substack{m', m \\ i(m', m) = 0}} \left(I \alpha_{t-1}(m') + I \gamma_t(m', m) + I \beta_t(m) \right) \end{aligned} \quad (31)$$

【0102】

このような復号装置3は、受信装置により受信された軟入力の受信値 y_t を入力すると、 $I \gamma$ 算出・記憶回路3 2によって、受信値 y_t を受信する毎に、対数尤度 $I \gamma_t(m', m)$ を算出し、 $I \alpha$ 算出・記憶回路3 3によって、対数尤度 $I \alpha_t(m)$ を算出した後、全ての受信値 y_t を受信すると、 $I \beta$ 算出・記憶回路3 4によって、全ての時刻 t における各ステート m について、対数尤度 $I \beta_t(m)$ を算出する。そして、復号装置3は、軟出力算出回路3 5によって、算出した対数尤度 $I \alpha_t$, $I \beta_t$ 及び $I \gamma_t$ を用いて、各時刻 t における対数軟出力 $I \lambda_t$ を算出する。このように、復号装置3は、 $\text{Log}-\text{BCJR}$ アルゴリズムを適用した軟出力復号を行うことができる。

【0103】

さて、復号装置3は、 $I \alpha$ 算出・記憶回路3 3 及び $I \beta$ 算出・記憶回路3 4 における対数尤度 $I \alpha_t$, $I \beta_t$ を算出する回路規模を削減したものである。そこで、以下では、 $I \alpha$ 算出・記憶回路3 3 及び $I \beta$ 算出・記憶回路3 4 について、より具体的に詳述する。

【0104】

まず、 $I \alpha$ 算出・記憶回路3 3 について説明する。 $I \alpha$ 算出・記憶回路3 3 は

、図5に示すように、算出された対数尤度 $I \alpha$ と対数尤度の初期値 $I \alpha_0$ とのいずれか一方を選択するセレクタ4 1と、初期値 $I \alpha_0$ 又は対数尤度 $I \alpha$ を保持するレジスタ4 2と、各ステートにおける対数尤度 $I \alpha$ を算出する $I \alpha$ 算出回路4 3と、各ステートにおける対数尤度 $I \alpha$ を順次保持するRAM(Random Access Memory)4 4, 4 5と、これらのRAM4 4, 4 5から読み出した対数尤度 $I \alpha$ を選択的に取り出す選択回路4 6とを有する。

【0105】

セレクタ4 1は、コントローラ3 1から供給されたコントロール信号SC α による制御の下に、初期化時には対数尤度の初期値 $I \alpha_0$ を選択し、初期化時以外の時には $I \alpha$ 算出回路4 3から供給される対数尤度 $I \alpha$ を選択する。なお、初期化は、 $I \gamma$ 算出・記憶回路3 2からの対数尤度 $I \gamma(\alpha)$ の出力が開始される1時刻前の時点に行われる。ここで、復号装置3が符号化装置1による符号化の開始時点を把握している場合には、初期値 $I \alpha_0$ としては、ステート0における値として $\log 1 = 0$ が与えられ、その他のステートにおける値として $\log 0 = -\infty$ が与えられる。一方、復号装置3が符号化装置1による符号化の開始時点を把握していない場合には、初期値 $I \alpha_0$ としては、全てのステートに対して $\log(1/M)$ 、ここでは $\log(1/4)$ が与えられるが、実際には、全てのステートに対して同じ値が与えられればよく、例えば全てのステートに対して0が与えられてもよい。セレクタ4 1は、初期値 $I \alpha_0$ 又は対数尤度 $I \alpha$ のうちの選択した一方をレジスタ4 2に供給する。

【0106】

レジスタ4 2は、セレクタ4 1から供給される初期値 $I \alpha_0$ 又は対数尤度 $I \alpha$ を保持する。そして、レジスタ4 2は、次時刻において、保持している初期値 $I \alpha_0$ 又は対数尤度 $I \alpha$ を $I \alpha$ 算出回路4 3及びRAM4 4, 4 5に供給する。

【0107】

$I \alpha$ 算出回路4 3は、図6に示すように、各ステートに応じた数、ここでは4つの加算比較選択回路4 7₀, 4 7₁, 4 7₂, 4 7₃を有する。

【0108】

加算比較選択回路4 7₀, 4 7₁, 4 7₂, 4 7₃には、それぞれ、トレス上の

遷移に基づいて、 $I\gamma$ 算出・記憶回路32により算出されたトレリス上の出力“00”、“10”、“01”、“11”に対応する枝の対数尤度 $I\gamma_t[00]$ 、 $I\gamma_t[10]$ 、 $I\gamma_t[01]$ 、 $I\gamma_t[11]$ と、各ステートにおける1時刻前の対数尤度 $I\alpha_{t-1}(0)$ 、 $I\alpha_{t-1}(1)$ 、 $I\alpha_{t-1}(2)$ 、 $I\alpha_{t-1}(3)$ が供給される。そして、加算比較選択回路47₀、47₁、47₂、47₃は、それぞれ、次時刻のステート0、ステート1、ステート2、ステート3における対数尤度 $I\alpha$ を求める。

【0109】

具体的には、加算比較選択回路47₀は、対数尤度 $I\gamma_t[00]$ 、 $I\gamma_t[11]$ を入力するとともに、対数尤度 $I\alpha_{t-1}(0)$ 、 $I\alpha_{t-1}(2)$ を入力し、ステート0における対数尤度 $I\alpha_t(0)$ を求める。

【0110】

また、加算比較選択回路47₁は、対数尤度 $I\gamma_t[11]$ 、 $I\gamma_t[00]$ を入力するとともに、対数尤度 $I\alpha_{t-1}(0)$ 、 $I\alpha_{t-1}(2)$ を入力し、ステート1における対数尤度 $I\alpha_t(1)$ を求める。

【0111】

さらに、加算比較選択回路47₂は、対数尤度 $I\gamma_t[10]$ 、 $I\gamma_t[01]$ を入力するとともに、対数尤度 $I\alpha_{t-1}(1)$ 、 $I\alpha_{t-1}(3)$ を入力し、ステート2における対数尤度 $I\alpha_t(2)$ を求める。

【0112】

さらにまた、加算比較選択回路47₃は、対数尤度 $I\gamma_t[01]$ 、 $I\gamma_t[10]$ を入力するとともに、対数尤度 $I\alpha_{t-1}(1)$ 、 $I\alpha_{t-1}(3)$ を入力し、ステート3における対数尤度 $I\alpha_t(3)$ を求める。

【0113】

このような $I\alpha$ 算出回路43は、 $I\gamma$ 算出・記憶回路32から供給された対数尤度 $I\gamma(\alpha)$ と、レジスタ42に保持されていた1時刻前の初期値 $I\alpha_0$ 又は対数尤度 $I\alpha$ とを用いて、上式(27)に示した演算、すなわち、上式(28)に示した演算を行い、次時刻の各ステートにおける対数尤度 $I\alpha$ を算出する。 $I\alpha$ 算出回路43は、算出した対数尤度 $I\alpha$ をセレクタ41に供給する。なお、加

算比較選択回路47₀, 47₁, 47₂, 47₃については、後に詳述する。

【0114】

RAM44, 45は、それぞれ、コントローラ31から供給されたコントロール信号SC α による制御の下に、レジスタ42から供給された対数尤度I α (0), I α (1), I α (2), I α (3)を順次保持する。ここで、対数尤度I α (0), I α (1), I α (2), I α (3)のビット数を、それぞれ、例えば8ビットとすると、RAM44, 45は、それぞれ、32ビットを1ワードとして、対数尤度I α (0), I α (1), I α (2), I α (3)を保持する。これらのRAM44, 45に保持された対数尤度I α (0), I α (1), I α (2), I α (3)は、選択回路46により所定の順序で読み出される。

【0115】

選択回路46は、コントローラ31から供給されたコントロール信号SC α による制御の下に、RAM44, 45から読み出した対数尤度I α (0), I α (1), I α (2), I α (3)を選択的に取り出し、対数尤度I α (λ)として軟出力算出回路35に供給する。

【0116】

このようなI α 算出・記憶回路33は、I γ 算出・記憶回路32からの対数尤度I γ (α)の出力が開始される1時刻前の時点で初期化を行い、セレクタ41により選択された初期値I α_0 をレジスタ42に保持させる。そして、I α 算出・記憶回路33は、以後のクロック周期において、I α 算出回路43によって、I γ 算出・記憶回路32から供給された対数尤度I γ (α)と、レジスタ42から供給された1時刻前の対数尤度I α とを用いて、次時刻における対数尤度I α を順次算出し、その対数尤度I α をレジスタ42に新たに保持させる。また、I α 算出・記憶回路33は、レジスタ42に保持された各ステートにおける対数尤度I α (0), I α (1), I α (2), I α (3)を順次RAM44, 45に保持させるとともに、選択回路46により所定の順序で読み出し、対数尤度I α (λ)として軟出力算出回路35に供給する。

【0117】

つぎに、I β 算出・記憶回路34について説明する。I β 算出・記憶回路34

は、図7に示すように、各ステートにおける対数尤度 $I\beta$ を算出する $I\beta$ 算出回路 $51_1, 51_2$ と、算出された対数尤度 $I\beta$ と対数尤度の初期値 $I\beta_a, I\beta_b$ とのいずれか一方を選択するセレクタ $52_1, 52_2$ と、初期値 $I\beta_a, I\beta_b$ 又は対数尤度 $I\beta$ を保持するレジスタ $53_1, 53_2$ と、これらのレジスタ $53_1, 53_2$ から供給された対数尤度 $I\beta$ を選択的に取り出す選択回路 54 とを有する。

【0118】

$I\beta$ 算出回路 $51_1, 51_2$ は、それぞれ、図8に示すように、各ステートに応じた数、ここでは4つの加算比較選択回路 $55_0, 55_1, 55_2, 55_3$ を有する。

【0119】

加算比較選択回路 $55_0, 55_1, 55_2, 55_3$ には、それぞれ、トレリス上の遷移に基づいて、 $I\gamma$ 算出・記憶回路 32 により算出されたトレリス上の出力“00”、“10”、“01”、“11”に対応する枝の対数尤度 $I\gamma_t[00], I\gamma_t[10], I\gamma_t[01], I\gamma_t[11]$ と、各ステートにおける対数尤度 $I\beta_t(0), I\beta_t(1), I\beta_t(2), I\beta_t(3)$ が供給される。そして、加算比較選択回路 $55_0, 55_1, 55_2, 55_3$ は、それぞれ、1時刻前のステート0、ステート1、ステート2、ステート3における対数尤度 $I\beta$ を求める。

【0120】

具体的には、加算比較選択回路 55_0 は、対数尤度 $I\gamma_t[00], I\gamma_t[11]$ を入力するとともに、対数尤度 $I\beta_t(0), I\beta_t(1)$ を入力し、ステート0における対数尤度 $I\beta_{t-1}(0)$ を求める。

【0121】

また、加算比較選択回路 55_1 は、対数尤度 $I\gamma_t[10], I\gamma_t[01]$ を入力するとともに、対数尤度 $I\beta_t(2), I\beta_t(3)$ を入力し、ステート1における対数尤度 $I\beta_{t-1}(1)$ を求める。

【0122】

さらに、加算比較選択回路 55_2 は、対数尤度 $I\gamma_t[11], I\gamma_t[00]$

を入力するとともに、対数尤度 $I\beta_t(0)$, $I\beta_t(1)$ を入力し、ステート2における対数尤度 $I\beta_{t-1}(2)$ を求める。

【0123】

さらにまた、加算比較選択回路55₃は、対数尤度 $I\gamma_t[01]$, $I\gamma_t[10]$ を入力するとともに、対数尤度 $I\beta_t(2)$, $I\beta_t(3)$ を入力し、ステート3における対数尤度 $I\beta_{t-1}(3)$ を求める。

【0124】

このような $I\beta$ 算出回路51₁, 51₂は、それぞれ、 $I\gamma$ 算出・記憶回路32から供給された対数尤度 $I\gamma(\beta 1)$, $I\gamma(\beta 2)$ と、レジスタ53₁, 53₂に保持されていた初期値 $I\beta_a$, $I\beta_b$ 又は対数尤度 $I\beta$ を用いて、上式(29)に示した演算、すなわち、上式(30)に示した演算を行い、1時刻前の各ステートにおける対数尤度 $I\beta$ を算出する。ここで、対数尤度 $I\beta(0)$, $I\beta(1)$, $I\beta(2)$, $I\beta(3)$ のビット数は、それぞれ、例えば8ビットであり、総ビット数は、32ビットとなる。 $I\beta$ 算出回路51₁, 51₂は、それぞれ算出した対数尤度 $I\beta$ をセレクタ52₁, 52₂に供給する。なお、加算比較選択回路55₀, 55₁, 55₂, 55₃については、後に詳述する。

【0125】

セレクタ52₁, 52₂は、それぞれ、コントローラ31から供給されたコントロール信号 $SC\beta$ による制御の下に、初期化時には対数尤度の初期値 $I\beta_a$, $I\beta_b$ を選択し、初期化時以外の時には $I\beta$ 算出回路51₁, 51₂のそれぞれから供給される対数尤度 $I\beta$ を選択する。なお、初期化は、 $I\gamma$ 算出・記憶回路32からの対数尤度 $I\gamma(\beta 1)$, $I\gamma(\beta 2)$ の出力が開始される1時刻前の時点で行われ、以後打ち切り長の2倍の長さの周期毎に行われる。ここで、初期値 $I\beta_a$, $I\beta_b$ としては、通常、全てのステートに対して例えば0や $\log(1/M)$ 、ここでは $\log(1/4)$ といったように、同じ値が与えられるが、終結された符号を復号する際には、終結するステートにおける値として $\log 1 = 0$ が与えられ、その他のステートにおける値として $\log 0 = -\infty$ が与えられる。セレクタ52₁, 52₂は、それぞれ、初期値 $I\beta_a$, $I\beta_b$ 又は対数尤度 $I\beta$ のうちの選択した一方をレジスタ53₁, 53₂に供給する。

【0126】

レジスタ 53_1 , 53_2 は、それぞれ、セレクタ 52_1 , 52_2 から供給される初期値 $I\beta_a$, $I\beta_b$ 又は対数尤度 $I\beta$ を保持する。そして、レジスタ 53_1 , 53_2 は、それぞれ、次時刻において、保持している初期値 $I\beta_a$, $I\beta_b$ 又は対数尤度 $I\beta$ を $I\beta$ 算出回路 51_1 , 51_2 及び選択回路 54 に供給する。

【0127】

選択回路 54 は、コントローラ 31 から供給されたコントロール信号 $SC\beta$ による制御の下に、レジスタ 53_1 , 53_2 のそれぞれから供給された対数尤度 $I\beta(0)$, $I\beta(1)$, $I\beta(2)$, $I\beta(3)$ を選択的に取り出し、対数尤度 $I\beta(\lambda)$ として軟出力算出回路 35 に供給する。

【0128】

このような $I\beta$ 算出・記憶回路 34 は、 $I\gamma$ 算出・記憶回路 32 からの対数尤度 $I\gamma(\beta1)$ の出力が開始される1時刻前の時点及び以後打ち切り長の2倍の長さの周期毎に初期化を行い、セレクタ 52_1 により選択された初期値 $I\beta_a$ をレジスタ 53_1 に保持させる。そして、 $I\beta$ 算出・記憶回路 34 は、以後のクロック周期において、 $I\beta$ 算出回路 51_1 によって、 $I\gamma$ 算出・記憶回路 32 から供給された対数尤度 $I\gamma(\beta1)$ と、レジスタ 52_1 から供給された対数尤度 $I\beta$ とを用いて、1時刻前における対数尤度 $I\beta$ を順次算出し、その対数尤度 $I\beta$ をレジスタ 53_1 に新たに保持させる。

【0129】

また、 $I\beta$ 算出・記憶回路 34 は、 $I\gamma$ 算出・記憶回路 32 からの対数尤度 $I\gamma(\beta2)$ の出力が開始される1時刻前の時点及び以後打ち切り長の2倍の長さの周期毎に初期化を行い、セレクタ 52_2 により選択された初期値 $I\beta_b$ をレジスタ 53_2 に保持させる。そして、 $I\beta$ 算出・記憶回路 34 は、以後のクロック周期において、 $I\beta$ 算出回路 51_2 によって、 $I\gamma$ 算出・記憶回路 32 から供給された対数尤度 $I\gamma(\beta2)$ と、レジスタ 52_2 から供給された対数尤度 $I\beta$ とを用いて、1時刻前における対数尤度 $I\beta$ を順次算出し、その対数尤度 $I\beta$ をレジスタ 53_2 に新たに保持させる。そして、 $I\beta$ 算出・記憶回路 34 は、レジスタ 53_1 , 53_2 のそれぞれに保持された各ステートにおける対数尤度 $I\beta(0)$

, $I\beta(1)$, $I\beta(2)$, $I\beta(3)$ を選択回路54により所定の順序で読み出し、対数尤度 $I\beta(\lambda)$ として軟出力算出回路35に供給する。

【0130】

さて、 $I\alpha$ 算出・記憶回路33が有する加算比較選択回路 47_0 , 47_1 , 47_2 , 47_3 と、 $I\beta$ 算出・記憶回路34が有する加算比較選択回路 55_0 , 55_1 , 55_2 , 55_3 について説明するが、これらの加算比較選択回路 47_0 , 47_1 , 47_2 , 47_3 , 55_0 , 55_1 , 55_2 , 55_3 は、それぞれ、入力と出力が異なるものの同一の構成からなる。そこで、以下では、加算比較選択回路 47_0 , 47_1 , 47_2 , 47_3 , 55_0 , 55_1 , 55_2 , 55_3 を、加算比較選択回路60と総称して説明する。また、以下では、加算比較選択回路 47_0 , 47_1 , 47_2 , 47_3 のそれぞれに入力される2つの対数尤度 $I\gamma$ 及び加算比較選択回路 55_0 , 55_1 , 55_2 , 55_3 のそれぞれに入力される2つの対数尤度 $I\gamma$ を、IA, IBと総称し、加算比較選択回路 47_0 , 47_1 , 47_2 , 47_3 のそれぞれに入力される2つの対数尤度 $I\alpha$ 及び加算比較選択回路 55_0 , 55_1 , 55_2 , 55_3 のそれぞれに入力される2つの対数尤度 $I\beta$ を、IC, IDと総称し、加算比較選択回路 47_0 , 47_1 , 47_2 , 47_3 のそれぞれから出力される対数尤度 $I\alpha$ 及び加算比較選択回路 55_0 , 55_1 , 55_2 , 55_3 のそれぞれから出力される対数尤度 $I\beta$ を、IEと総称して説明する。なお、以下の説明では、ハードウェアとして実装することを考慮して、確率を0以上の値で表し、確率が低いものほど大きな値で表すものとする。

【0131】

加算比較選択回路60は、図9に示すように、2つのデータを加算する加算器61, 62と、これらの加算器61, 62からの出力の大小を比較する比較回路63と、加算器61, 62のそれぞれからの出力のいずれか一方を選択するセレクタ64と、Log-BCJRアルゴリズムにおける補正項の値を算出する補正項算出回路65と、2つのデータの差分をとる差分器66とを有する。

【0132】

加算器61は、対数尤度IA, ICを入力し、これらの対数尤度IA, ICを加算する。例えば、加算比較選択回路60が加算比較選択回路 47_0 の場合には

、加算器61は、対数尤度 $I_{\gamma_t}[00]$ ，対数尤度 $I_{\alpha_{t-1}}(0)$ を入力し、これらの対数尤度 $I_{\gamma_t}[00]$ ，対数尤度 $I_{\alpha_{t-1}}(0)$ を加算する。加算器61は、加算して得られたデータを比較回路63、セレクタ64及び補正項算出回路65に供給する。なお、以下では、加算器61から出力されるデータをPとして説明する。

【0133】

加算器62は、対数尤度IB，IDを入力し、これらの対数尤度IB，IDを加算する。例えば、加算比較選択回路60が加算比較選択回路470の場合には、加算器62は、対数尤度 $I_{\gamma_t}[11]$ ，対数尤度 $I_{\alpha_{t-1}}(2)$ を入力し、これらの対数尤度 $I_{\gamma_t}[11]$ ，対数尤度 $I_{\alpha_{t-1}}(2)$ を加算する。加算器62は、加算して得られたデータを比較回路63、セレクタ64及び補正項算出回路65に供給する。なお、以下では、加算器62から出力されるデータをQとして説明する。

【0134】

比較回路63は、加算器61から供給されたデータPの値と、加算器62から供給されたデータQの値との大小を比較する。比較回路63は、比較結果を示す比較結果情報をセレクタ64に供給する。

【0135】

セレクタ64は、比較回路63から供給された比較結果情報に基づいて、加算器61から供給されたデータPと、加算器62から供給されたデータQとのうち、値が小さいもの、すなわち、確率が高いものを選択する。セレクタ64は、選択したデータを差分器66に供給する。なお、このセレクタ64により選択されたデータは、上式(28)又は上式(30)における右辺第1項を示すものに他ならない。

【0136】

補正項算出回路65は、加算器61から供給されたデータPと、加算器62から供給されたデータQとの差分値の絶対値を算出する絶対値算出回路67と、この絶対値算出回路67により算出された絶対値を用いて補正項を線形近似により算出する線形近似手段である線形近似回路68とを有する。補正項算出回路65

は、Log-BCJRアルゴリズムにおける補正項、すなわち、上式(28)又は上式(30)における右辺第2項の値を算出する。具体的には、補正項算出回路65は、補正項を、変数 $|P-Q|$ に対する1次元の関数で表し、この関数の傾きを表す係数 $-a$ ($a > 0$)と、関数の切片を表す係数 b とを用いて $-a|P-Q| + b$ の形に線形近似した値を算出する。補正項算出回路65は、算出して得られたデータ Z を差分器66に供給する。

【0137】

差分器66は、セレクタ64により選択されたデータと、補正項算出回路65から供給されたデータ Z との差分値を求め、この差分値を対数尤度IEとして出力する。例えば、加算比較選択回路60が加算比較選択回路470の場合には、差分器66は、対数尤度 $I\alpha_t(0)$ を出力する。

【0138】

以下では、このような加算比較選択回路60における遅延量を見積もることを考える。なお、比較回路による遅延量及び差分器による遅延量は、ともに、加算器61, 62のような通常の加算器による遅延量と同じものとする。

【0139】

加算比較選択回路60において回避できない遅延量は、同図から明らかなように、1つの加算器に相当するものとして、加算器61, 62による遅延量と、比較回路63による遅延量と、差分器66による遅延量とがある。すなわち、加算比較選択回路60は、少なくとも3つの加算器分の遅延量を有することになる。

【0140】

したがって、加算比較選択回路60における遅延量を可能な限り小さくするためにには、補正項算出回路65による遅延量を小さくする必要がある。そこで、補正項算出回路65による遅延量を見積もることを考える。

【0141】

まず、絶対値算出回路67による遅延量を見積もる。絶対値算出回路67は、例えば図10に示すように、加算器61, 62からの出力の大小を比較する比較回路71と、2つのデータの差分をとる2つの差分器72, 73と、これらの差分器72, 73のそれぞれからの出力のいずれか一方を選択するセレクタ74と

を有するものとして実装することができる。

【0142】

すなわち、絶対値算出回路67は、比較回路71によって、加算器61から供給されたデータPの値と、加算器62から供給されたデータQの値との大小を比較する。これと同時に、絶対値算出回路67は、差分器72によって、加算器61から供給されたデータPと、加算器62から供給されたデータQとの差分値($P - Q$)を求めるとともに、差分器73によって、加算器62から供給されたデータQと、加算器61から供給されたデータPとの差分値($Q - P$)を求める。そして、絶対値算出回路67は、セレクタ74によって、比較回路71による比較結果を示す比較結果情報に基づいて、差分値($P - Q$)と差分値($Q - P$)とのうち、正の値を有するものを選択し、選択した差分値を絶対値データ $|P - Q|$ として線形近似回路68に供給する。

【0143】

このような絶対値算出回路67は、比較回路71による処理と、差分器72, 73による処理とが並列的に行われることから、1つの加算器と1つのセレクタの遅延量を有するものと見積もることができる。

【0144】

つぎに、線形近似回路68による遅延量を見積もる。線形近似回路68は、図11に示すように、曲線Cに示す関数 $F = \log(1 + e^{-|P-Q|})$ を直線 L_1 , L_2 に示すような関数 $F = -a|P-Q| + b$ で近似するいわゆる線形近似法による $\log - sum$ 補正を行い、補正項の値を算出する。このとき、線形近似回路68は、少なくとも変数 $|P-Q|$ に乘算すべき関数 $F = -a|P-Q| + b$ の傾きを表す係数 $-a$ を2のべき乗を用いて表現して補正項の値を算出する。ここで、同図を参照すると、2のべき乗を用いて表現される係数 $-a$ としては、直線 L_1 に示すように、 $-a = -2^{-1} = -0.5$ とするか、若しくは、直線 L_2 に示すように、 $-a = -2^{-2} = -0.25$ とすることが考えられる。ここでは、 $-a = -0.25$ として説明する。この場合、線形近似回路68は、図12に示すように、差分器81と、比較回路82と、セレクタ83とを有するものとして実装することができる。

【0145】

差分器81は、関数 $F = -a |P - Q| + b$ の切片を表す係数 b と、絶対値算出回路67から供給された n ビットからなる絶対値データ $|P - Q|$ のうちの上位 $n - 2$ ビットとの差分値を求め、この差分値をセレクタ83に供給する。

【0146】

比較回路82は、係数 b の値と、絶対値データ $|P - Q|$ のうちの上位 $n - 2$ ビットで表されるデータ $|P - Q| [n : 3]$ の値との大小を比較する。比較回路82は、比較結果を示す比較結果情報をセレクタ83に供給する。

【0147】

セレクタ83は、比較回路82から供給された比較結果情報に基づいて、差分器81から供給されたデータと、“0”とのうち、いずれか一方を選択する。具体的には、セレクタ83は、比較回路82による比較の結果、 $|P - Q| [n : 3] \leq b$ であった場合には、差分器81から供給されたデータを選択し、 $|P - Q| [n : 3] > b$ であった場合には、“0”を選択する。セレクタ83は、選択したデータを補正項を示すデータZとして、差分器66に供給する。

【0148】

このような線形近似回路68は、絶対値算出回路67から供給された n ビットからなる絶対値データ $|P - Q|$ の下位1ビット目から下位2ビット目までを切り捨て、残りの上位 $n - 2$ ビットで表されるデータを係数 b から差分する。すなわち、線形近似回路68は、絶対値データ $|P - Q|$ のうちの下位2ビットを切り捨ててビットシフトすることによって、 $|P - Q|$ を $1/4 = 0.25$ 倍することができ、残りの上位 $n - 2$ ビットで表されるデータを係数 b から差分することによって、結果として “ $-0.25 |P - Q| + b$ ” の演算を実現することができる。

【0149】

また、線形近似回路68は、補正項が正值であることから、比較回路82による比較の結果、差分器81から出力されたデータの値が負であった場合、すなわち、補正項が負値として算出された場合には、セレクタ83により “0” を出力することによって、補正項が負値をとることを回避することができる。

【0150】

なお、例えば $-a = -2^{-1} = -0.5$ とした場合には、線形近似回路68は、絶対値データ $|P-Q|$ の下位1ビットを切り捨ててビットシフトすればよく、係数 $-a$ を表現するべき数に応じて、絶対値データ $|P-Q|$ の下位ビットから切り捨てるようすればよい。

【0151】

このような線形近似回路68は、実際の乗算器を不要とし、差分器81による処理と、比較回路82による処理とが並列的に行われることから、1つの加算器と1つのセレクタ分の遅延量を有するものと見積もることができる。

【0152】

さらに、線形近似回路68としては、係数 b をも2のべき乗を用いて表現して補正項の値を算出するものも考えられる。すなわち、係数 b を $2^m - 1$ で表現される値とし、 $-a = -0.25$ とした場合、線形近似回路68は、図13に示すように、インバータ91と、ORゲート92と、セレクタ93とを有するものとして実装することができる。

【0153】

インバータ91は、絶対値算出回路67から供給されたnビットからなる絶対値データ $|P-Q|$ のうちの下位3ビット目から下位 $m+2$ ビット目までのmビットを反転する。インバータ91は、反転して得られたデータをセレクタ93に供給する。

【0154】

ORゲート92は、絶対値算出回路67から供給されたnビットからなる絶対値データ $|P-Q|$ のうちの下位 $m+3$ ビット目からnビット目までの上位 $n-m-2$ ビットの論理和をとる。ORゲート92は、求めた論理和をセレクタ93に供給する。

【0155】

セレクタ93は、ORゲート92から供給された論理和に基づいて、インバータ91から供給されたデータと、“0”とのうち、いずれか一方を選択する。具体的には、セレクタ93は、ORゲート92から供給された論理和が“0”であ

った場合には、インバータ91から供給されたデータを選択し、ORゲート92から供給された論理和が“1”であった場合には、“0”を選択する。セレクタ93は、選択したデータを補正項を示すデータ乙として、差分器66に供給する。

【0156】

このような線形近似回路68は、絶対値算出回路67から供給されたnビットからなる絶対値データ $|P-Q|$ の下位1ビット目から下位2ビット目までを切り捨て、残りの上位 $n-2$ ビットのうちの下位3ビット目から下位 $m+2$ ビット目までのmビットをインバータ91により反転する。これと同時に、線形近似回路68は、ORゲート92によって、下位 $m+3$ ビット目からnビット目までの $n-m-2$ ビットの論理和をとる。

【0157】

すなわち、線形近似回路68は、上述したように、絶対値データ $|P-Q|$ のうちの下位2ビットを切り捨ててビットシフトすることによって、 $|P-Q|$ を $1/4 = 0.25$ 倍することができる。したがって、線形近似回路68は、絶対値データ $|P-Q|$ のうちの上位 $n-2$ ビットで表されるデータ $|P-Q|_{[n:3]}$ 、すなわち、 $0.25|P-Q|$ を -1 倍し、この値に $2^m - 1$ で表現される係数bを加算すればよい。

【0158】

ここで、線形近似回路68による演算を論理式で表現するために、図14(A)に示すように、nビットからなる絶対値データ $|P-Q|$ の下位2ビットを切り捨てて得られる $0.25|P-Q|$ を $A = (A_n, A_{n-1}, \dots, A_{m+3}, A_m + 2, \dots, A_3)$ とし、残りの上位 $n-2$ ビットのうちの下位3ビット目から下位 $m+2$ ビット目までのmビットと、下位 $m+3$ ビット目からnビット目までの $n-m-2$ ビットとを、それぞれ、 A' 、 A'' として説明する。

【0159】

まず、線形近似回路68により求めるべき“ $-0.25|P-Q| + 2^m - 1 = -A + (2^m - 1)$ ”が負値をとる場合を考える。この場合、次式(32)に示す同値関係が成立する。すなわち、“ $-A + (2^m - 1)$ ”が負値をとる場合

には、 A'' が正値をとることになる。換言すれば、“ $-A + (2^m - 1)$ ”が負値をとる場合には、 A'' を構成する全てのビットの論理和が“1”となることになる。

【0160】

【数32】

$$\begin{aligned} -A + (2^m - 1) < 0 &\Leftrightarrow A > 2^m - 1 \\ &\Leftrightarrow A'' > 0 \end{aligned} \quad (32)$$

【0161】

一方、線形近似回路68により求めるべき “ $-0.25 | P - Q | + 2^m - 1 = -A + (2^m - 1)$ ” が0以上の値をとる場合を考える。この場合、上式(32)に示した同値関係より $A'' = 0$ であることから、次式(33)が成立する。

【0162】

【数33】

$$-A + (2^m - 1) = -A' + (2^m - 1) \quad (33)$$

【0163】

ここで、 $2^m - 1$ は、mビット全てが“1”であるデータであることに着目すると、“ $-A' + (2^m - 1)$ ”は、同図(B)に示すように、 A' の否定で表される。

【0164】

以上の議論から、線形近似回路68は、Aの下位mビットの否定を求めればよいことになる。したがって、線形近似回路68は、絶対値データ $|P-Q|$ のうちの下位3ビット目から下位 $m+2$ ビット目までのmビットで表されるデータ $|P-Q|_{[m+2:2]}$ をインバータ91により反転することによって、“-0.25 $|P-Q|+2^{m-1}$ ”の演算を実現することができる。

【0165】

また、線形近似回路68は、ORゲート92によって、絶対値データ $|P-Q|$ のうちの下位 $m+3$ ビット目からnビット目までの $n-m-2$ ビットで表されるデータ $|P-Q|_{[m+3:n]}$ の論理和をとることによって、“-0.25 $|P-Q|+2^{m-1}$ ”の値の正負を判断することができる。そのため、線形近似回路68は、補正項が正值であることから、ORゲート92による論理和が“1”であった場合、すなわち、補正項が負値として算出された場合には、セレクタ93により“0”を出力することによって、補正項が負値をとることを回避することができる。

【0166】

なお、線形近似回路68は、係数aを -2^{-k} で表現したとき、絶対値データ $|P-Q|$ の下位1ビット目から下位kビット目までを切り捨ててビットシフトし、下位 $k+1$ ビット目から下位 $m+k$ ビット目までのmビットを反転することになる。例として、 $n=5$ 、 $m=2$ の場合、すなわち、“-0.25 $|P-Q|+3$ ”の演算を行う場合において、絶対値データ $|P-Q|$ とデータZとの関係を求めると、次表1に示すようになる。なお、同表には、上述したA'の否定、すなわち、インバータ91から出力されるデータについても示している。

【0167】

【表1】

表1 絶対値データ |P-QI| とデータ Zとの関係

P-QI	\overline{A}	Z
31	11111	00
30	11110	00
29	11101	00
28	11100	00
27	11011	01
26	11010	01
25	11001	01
24	11000	01
23	10111	10
22	10110	10
21	10101	10
20	10100	10
19	10011	11
18	10010	11
17	10001	11
16	10000	11
15	01111	00
14	01110	00
13	01101	00
12	01100	00
11	01011	01
10	01010	01
9	01001	01
8	01000	01
7	00111	10
6	00110	10
5	00101	10
4	00100	10
3	00011	11
2	00010	11
1	00001	11
0	00000	11

【0168】

同表に示すように、線形近似回路68は、絶対値データ $|P-Q|$ が0乃至12の範囲では、絶対値データ $|P-Q|$ のうちの下位3ビット目から下位 $2+2=4$ ビット目までの2ビットで表されるデータ $|P-Q| [4:2]$ をインバータ91により反転したものをデータZとして出力し、絶対値データ $|P-Q|$ が13以上の範囲では、インバータ91からの出力が負となることから、0を出力する。

【0169】

このような線形近似回路68は、実際の乗算器及び加算器を不要とし、ビットシフトとインバータのみで構成することができ、1つのセレクタ分の遅延量を有するものと見積もることができる。

【0170】

以上のように、補正項算出回路65による遅延量は、線形近似回路68を図12に示すように構成することによって、2つの加算器と2つのセレクタ分として見積もることができ、線形近似回路68を図13に示すように構成することによって、1つの加算器と2つのセレクタ分にまで抑えることができる。

【0171】

したがって、加算比較選択回路60における遅延量は、比較回路63による処理と、絶対値算出回路67による処理とが並列的に行われることから、線形近似回路68を図12に示すように構成した場合には、4つの加算器と3つのセレクタ分となり、線形近似回路68を図13に示すように構成した場合には、3つの加算器と3つのセレクタ分にまで抑えることができる。すなわち、加算比較選択回路60は、元々少なくとも3つの加算器分の遅延量を有するが、補正項算出回路65による遅延の影響を殆ど受けることなく、対数尤度IEを求めることができる。

【0172】

以上説明したように、符号化装置1と復号装置3とを用いて構成されるデータ送受信システムは、復号装置3において、線形近似によるlog-sum補正を行う際に、補正項を迅速に求めることができることから、性能を劣化させること

なく、動作速度を向上させることができる。

【0173】

すなわち、これらの符号化装置1と復号装置3とを用いて構成されるデータ送受信システムは、高性能且つ高速に畳み込み符号の復号を実現するものであり、ユーザに高い信頼性及び利便性を提供することができるものである。

【0174】

なお、本発明は、上述した実施の形態に限定されるものではなく、例えば、符号化装置としては、畳み込み演算を行うものでなくてもよく、いかなる符号化率の符号化を行うものであってもよい。

【0175】

また、本発明は、いわゆる並列連接畳み込み符号、縦列連接畳み込み符号、ターボ符号化変調方式による符号又は縦列連接符号化変調方式による符号といったように、複数の要素符号を連接して構成される符号の復号を行う場合にも容易に適用できるものである。

【0176】

さらに、上述した実施の形態では、符号化装置及び復号装置をデータ送受信システムにおける送信装置及び受信装置に適用して説明したが、本発明は、例えばフロッピーディスク、CD-ROM又はMO（Magneto Optical）といった磁気、光又は光磁気ディスク等の記録媒体に対する記録及び／又は再生を行う記録及び／又は再生装置に適用することもできる。この場合、符号化装置により符号化されたデータは、無記憶通信路に等価とされる記録媒体に記録され、復号装置により復号されて再生される。

【0177】

以上のように、本発明は、その趣旨を逸脱しない範囲で適宜変更が可能であることはいうまでもない。

【0178】

【発明の効果】

以上詳細に説明したように、本発明にかかる復号装置は、軟入力とされる受信値に基づいて任意のステートを通過する確率を対数表記した対数尤度を求め、こ

の対数尤度を用いて復号を行う復号装置であって、対数尤度を与えるために追加され、変数に対する1次元の関数で表される補正項を線形近似により算出する線形近似手段を備え、この線形近似手段は、少なくとも変数に乗算すべき関数の傾きを表す係数を2のべき乗を用いて表現して補正項を算出する。

【0179】

したがって、本発明にかかる復号装置は、線形近似手段によって、2のべき乗を用いて表現された係数を有する関数で表される補正項を算出することによって、性能を劣化させることなく、高速化を図ることができる。

【0180】

また、本発明にかかる復号方法は、軟入力とされる受信値に基づいて任意のステートを通過する確率を対数表記した対数尤度を求め、この対数尤度を用いて復号を行う復号方法であって、対数尤度を与えるために追加され、変数に対する1次元の関数で表される補正項を線形近似により算出する線形近似工程を備え、この線形近似工程では、少なくとも変数に乗算すべき関数の傾きを表す係数を2のべき乗を用いて表現して補正項を算出する。

【0181】

したがって、本発明にかかる復号方法は、線形近似工程にて、2のべき乗を用いて表現された係数を有する関数で表される補正項を算出することによって、性能を劣化させることなく、高速化を図ることを可能とする。

【図面の簡単な説明】

【図1】

本発明の実施の形態として示すデータ送受信システムを適用する通信モデルの構成を説明するブロック図である。

【図2】

同データ送受信システムにおける符号化装置の構成を説明するブロック図である。

【図3】

符号化装置におけるトレリスを説明する図である。

【図4】

同データ送受信システムにおける復号装置の構成を説明するブロック図である

【図5】

復号装置が備える $I\alpha$ 算出・記憶回路の構成を説明するブロック図である。

【図6】

$I\alpha$ 算出・記憶回路が有する $I\alpha$ 算出回路の構成を説明するブロック図である

【図7】

復号装置が備える $I\beta$ 算出・記憶回路の構成を説明するブロック図である。

【図8】

$I\beta$ 算出・記憶回路が有する $I\beta$ 算出回路の構成を説明するブロック図である

【図9】

$I\alpha$ 算出回路又は $I\beta$ 算出回路が有する加算比較選択回路の構成を説明するブロック図である。

【図10】

加算比較選択回路が有する絶対値算出回路の構成を説明するブロック図である

【図11】

加算比較選択回路が有する線形近似回路による $\log - \sin$ 補正を説明する図であって、補正項を示す関数と線形近似した関数とを示すグラフである。

【図12】

線形近似回路の構成を説明するブロック図であって、関数 $F = -a | P - Q | + b$ の係数 $-a$ を 2 のべき乗を用いて表現して補正項の値を算出する線形近似回路の構成を説明するブロック図である。

【図13】

図12に示す線形近似回路とは異なる他の線形近似回路の構成を説明するブロック図であって、関数 $F = -a | P - Q | + b$ の係数 $-a$ と係数 b とを 2 のべき乗を用いて表現して補正項の値を算出する線形近似回路の構成を説明するブロック図である。

ク図である。

【図14】

図13に示す線形近似回路による演算を説明する図である。

【図15】

通信モデルの構成を説明するブロック図である。

【図16】

従来の符号化装置におけるトレリスを説明する図であって、確率 α_t , β_t 及び γ_t の内容を説明するための図である。

【図17】

従来の復号装置において、BCJRアルゴリズムを適用して軟出力復号を行う際の一連の工程を説明するフローチャートである。

【図18】

従来の復号装置において、Max-Log-BCJRアルゴリズムを適用して軟出力復号を行う際の一連の工程を説明するフローチャートである。

【図19】

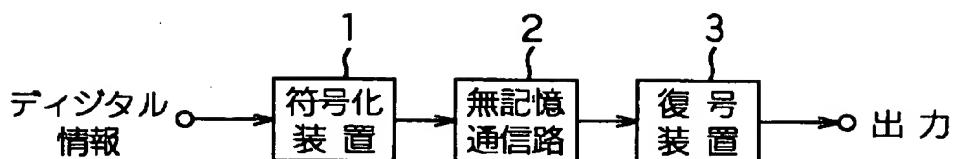
Log-Sum補正を説明する図であって、(A)は、補正項を示す関数と線形近似法により近似した関数とを示すグラフであり、(B)は、補正項を示す関数と閾値近似法により近似した関数とを示すグラフである。

【符号の説明】

1 符号化装置、 3 復号装置、 32 I γ 算出・記憶回路、 33 I α 算出・記憶回路、 34 I β 算出・記憶回路、 35 軟出力算出回路、
 43 I α 算出回路、 47₀, 47₁, 47₂, 47₃, 55₀, 55₁, 55₂,
 55₃, 60 加算比較選択回路、 51₁, 51₂ I β 算出回路、 65 補正項算出回路、 67 絶対値算出回路、 68 線形近似回路

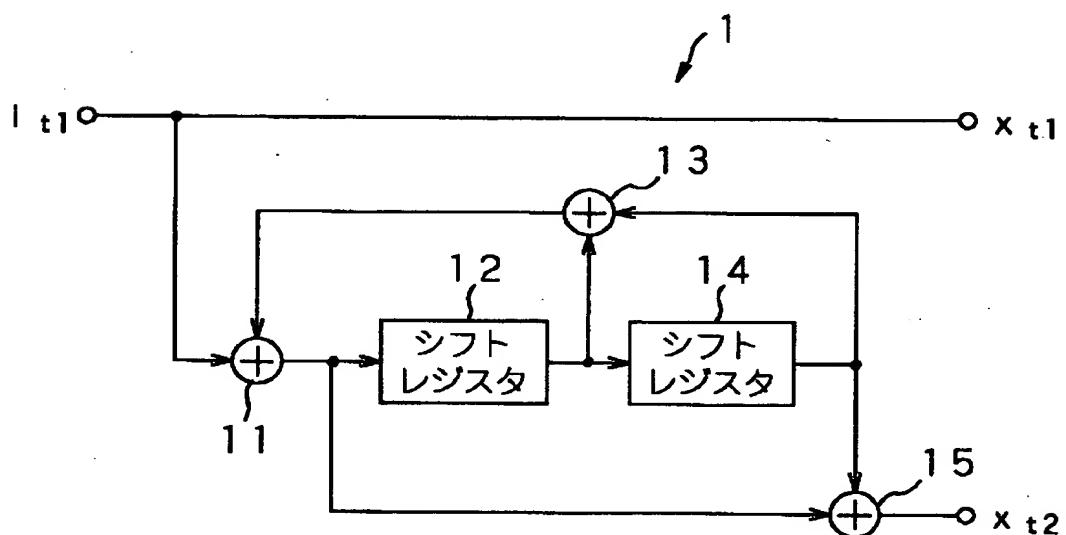
【書類名】 図面

【図1】



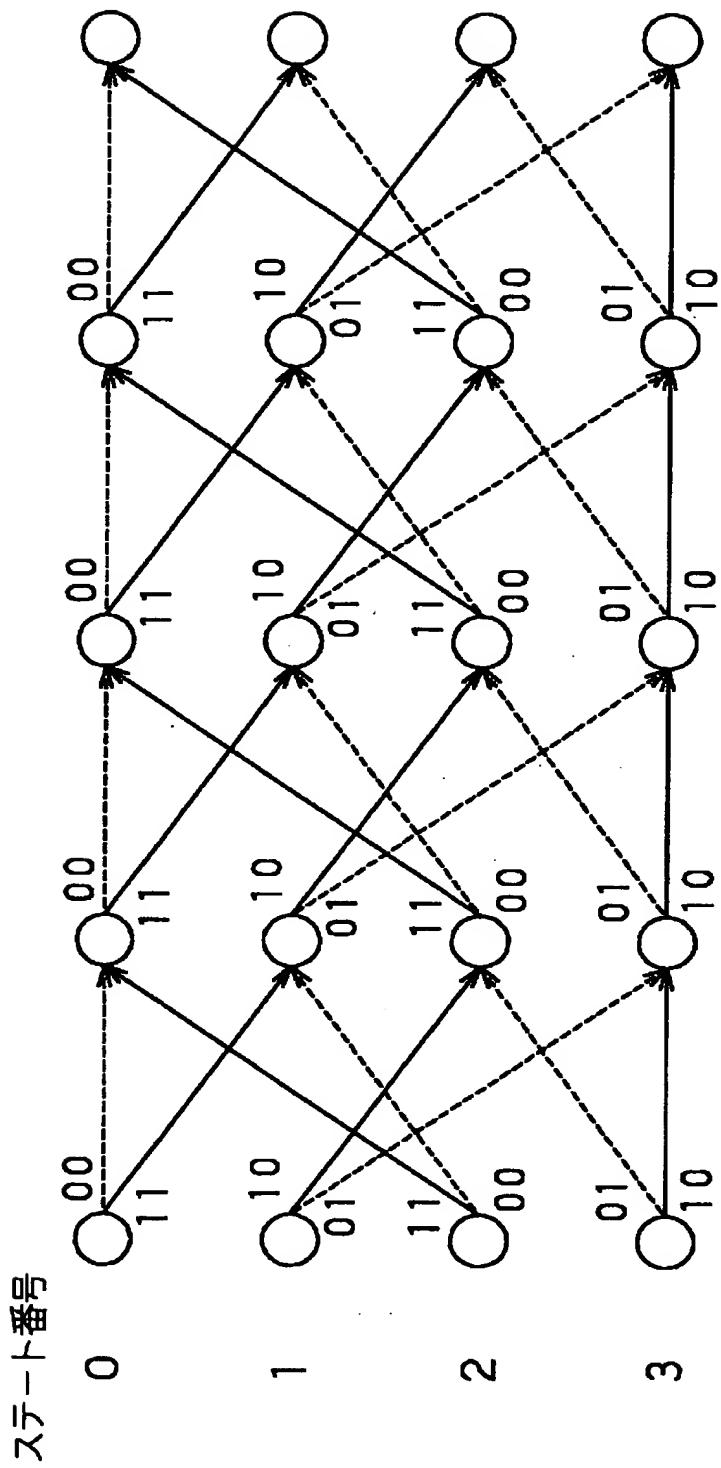
通信モデルの構成ブロック図

【図2】



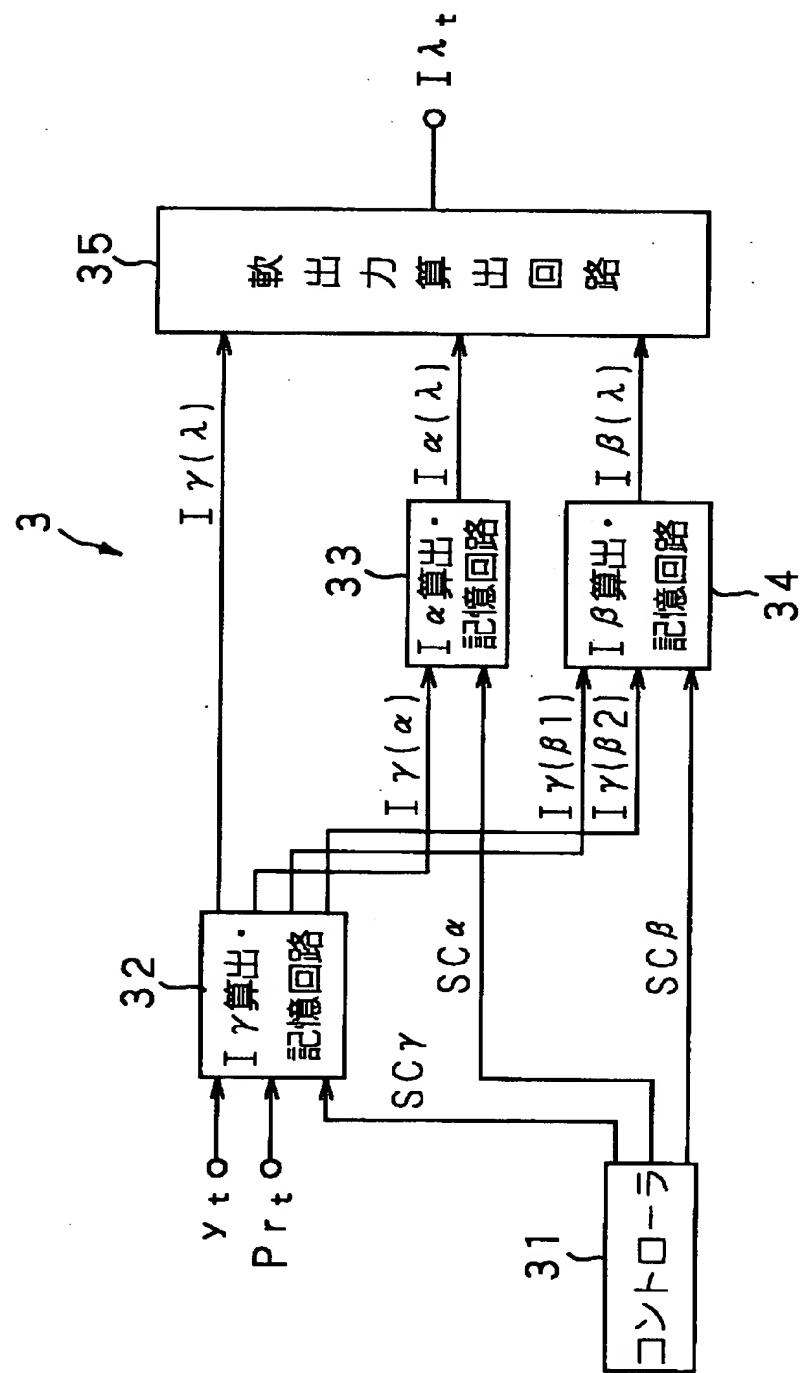
符号化装置の構成ブロック図

【図3】



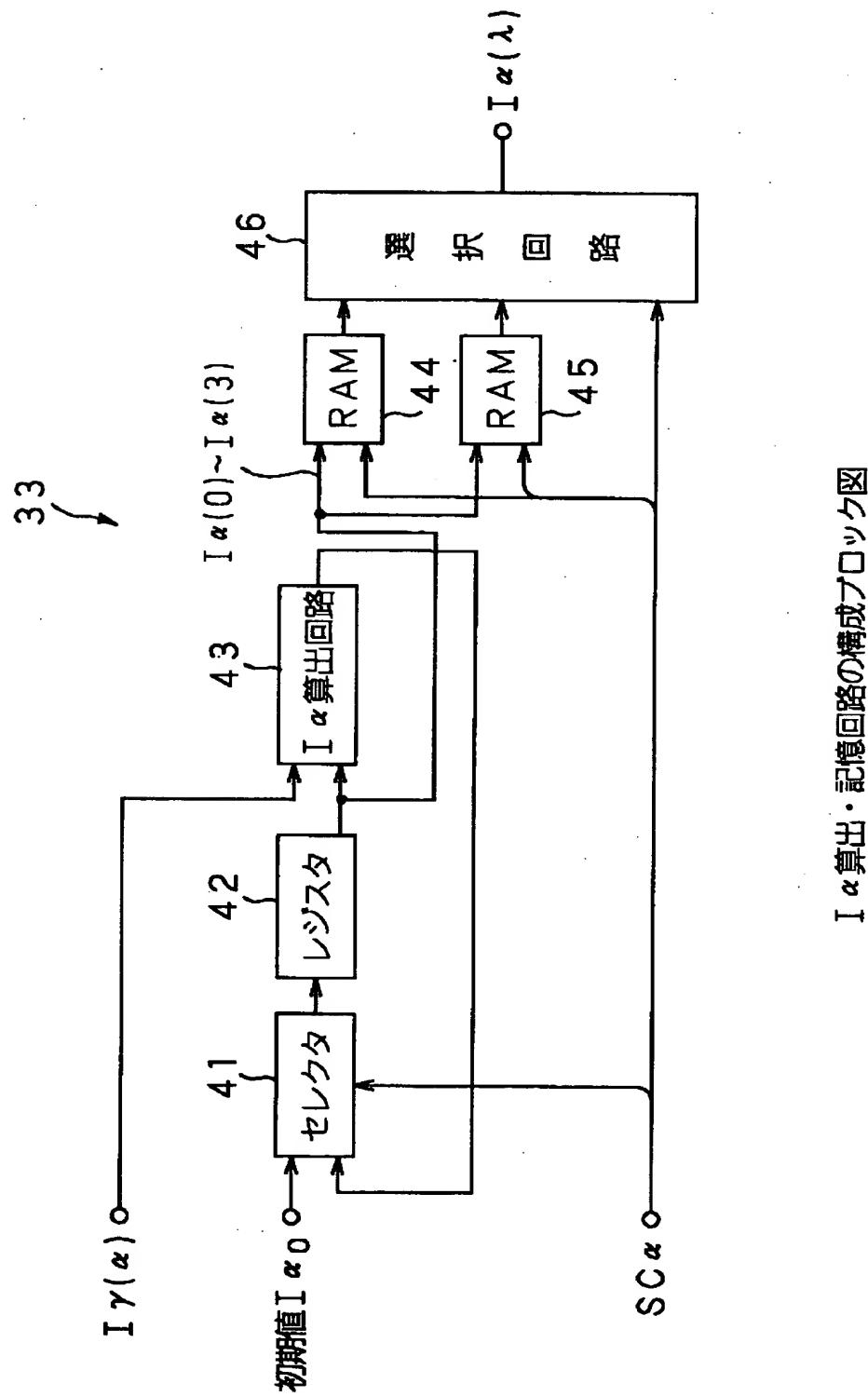
トレスの説明図

【図4】

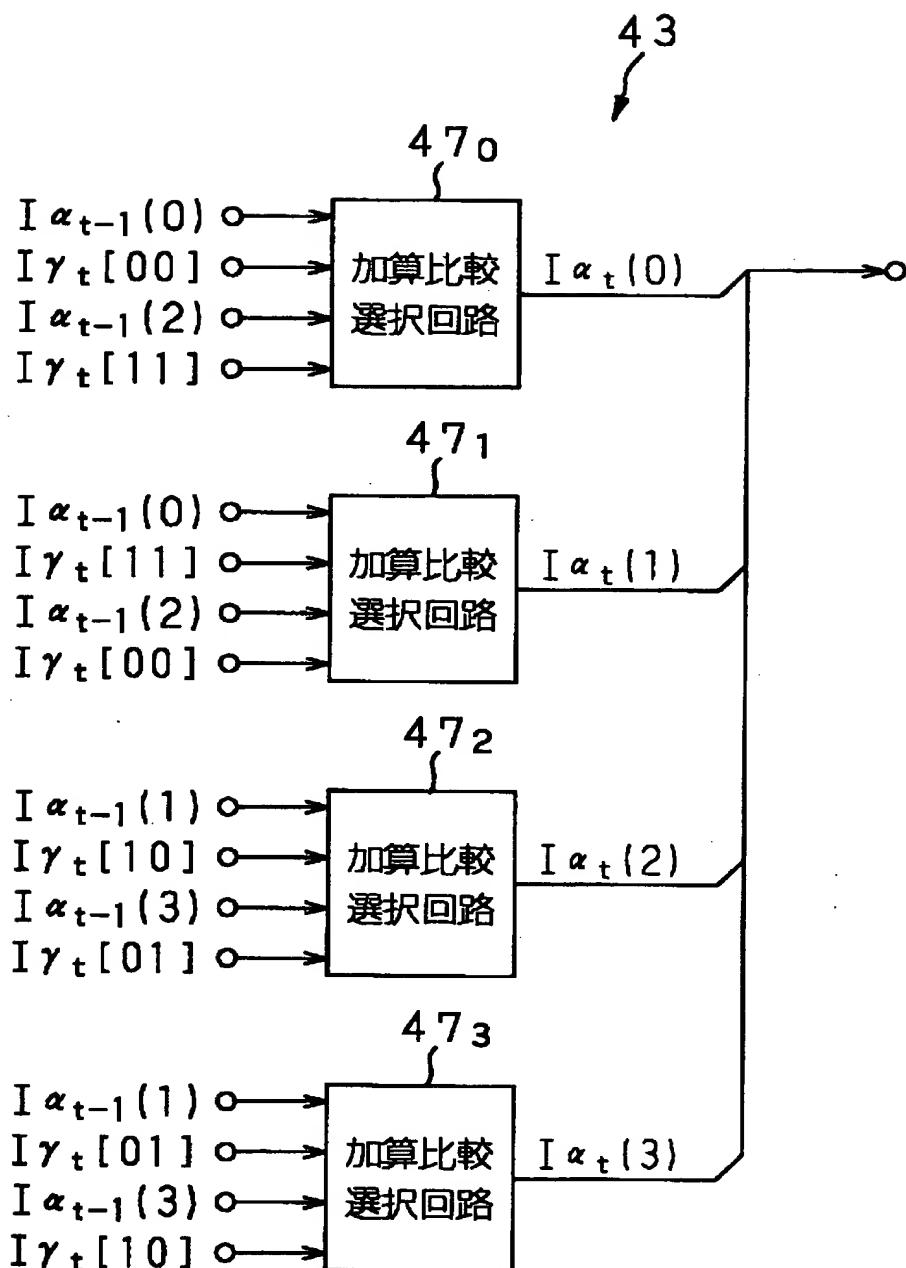


復号装置の構成ブロック図

【図5】

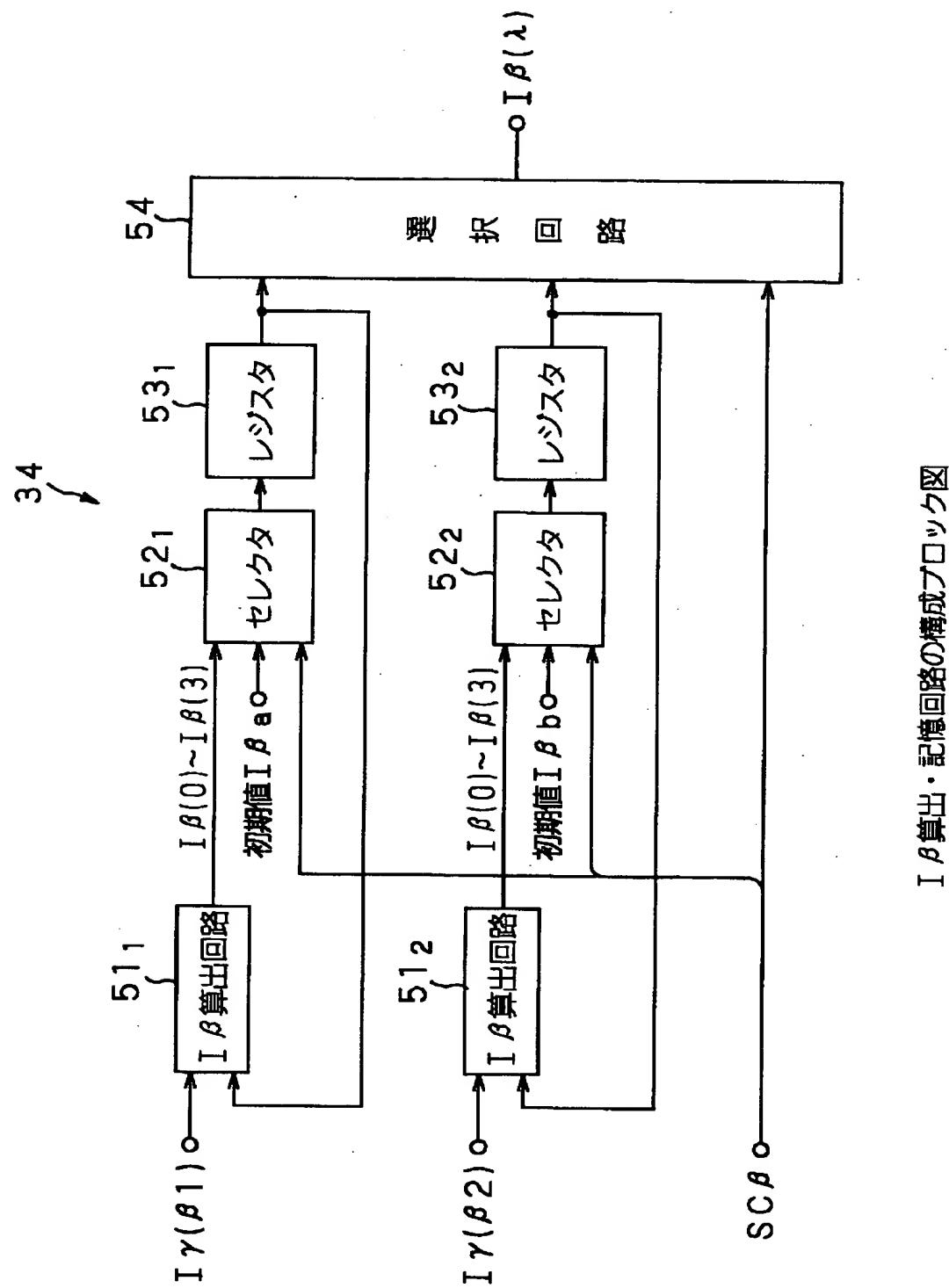
 $I\alpha$ 算出・記憶回路の構成ブロック図

【図6】



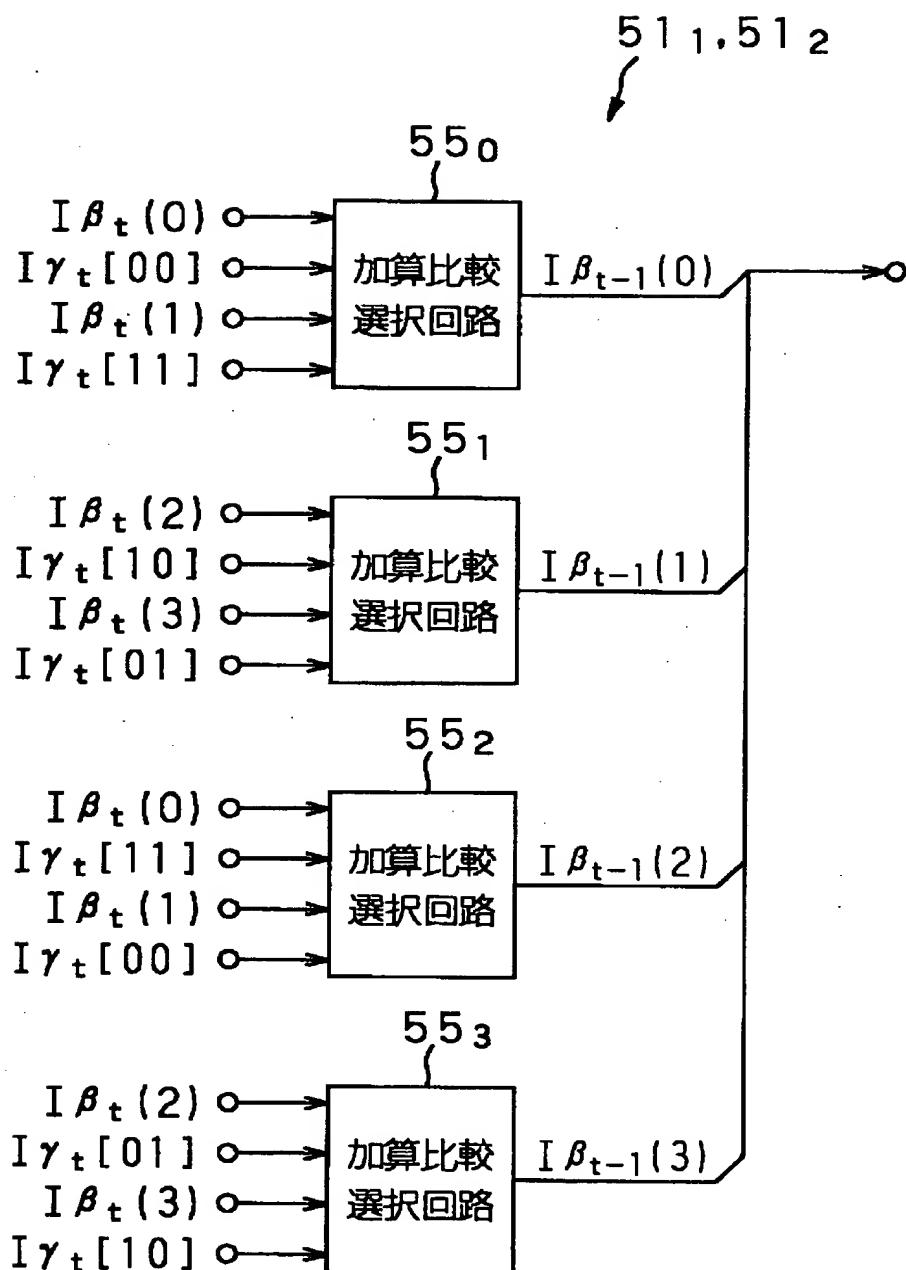
I-alpha算出回路の構成ブロック図

【図7】

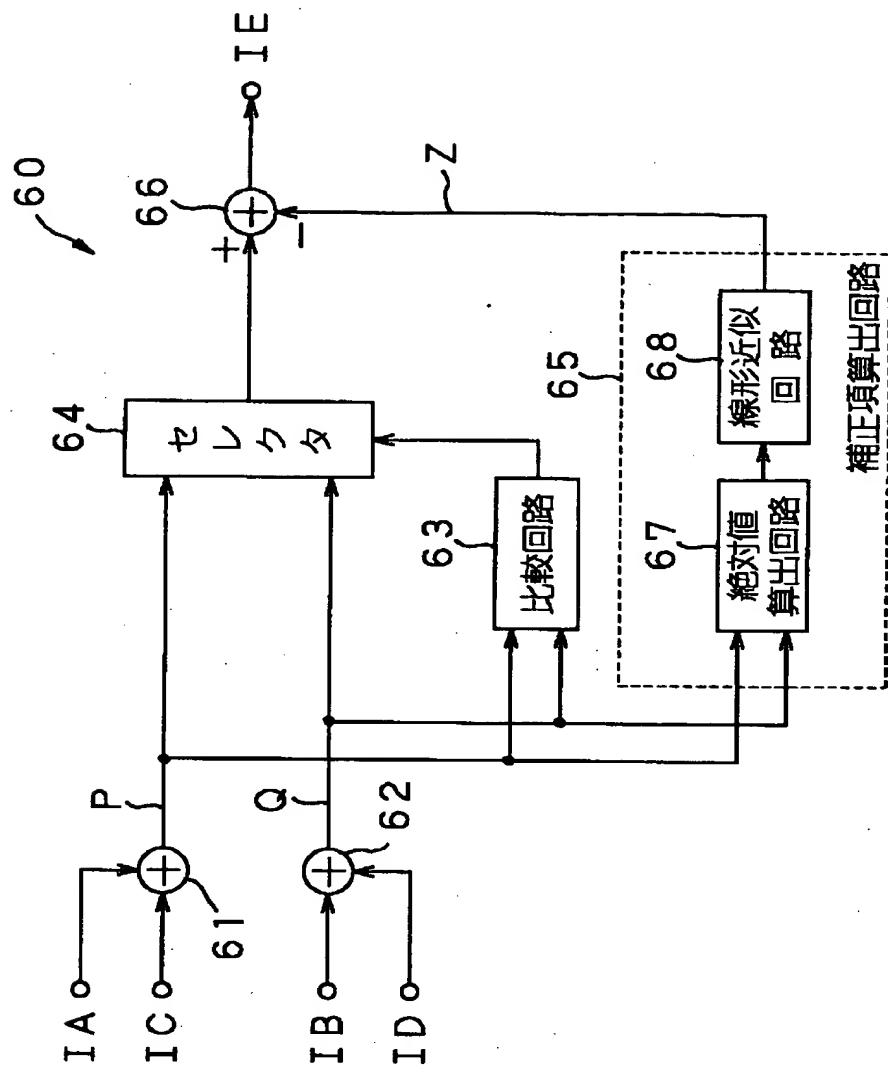


Iβ算出・記憶回路の構成ブロック図

【図8】

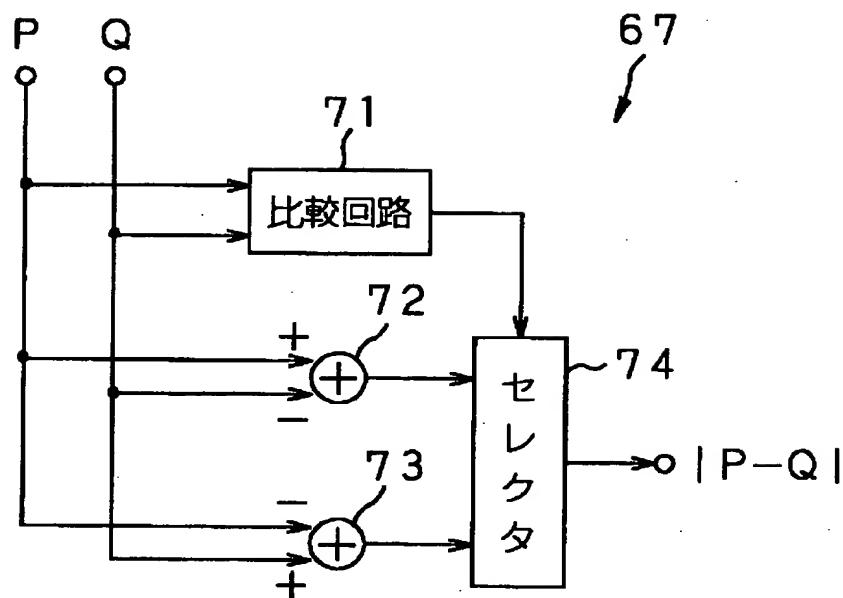
I β 算出回路の構成ブロック図

【図9】



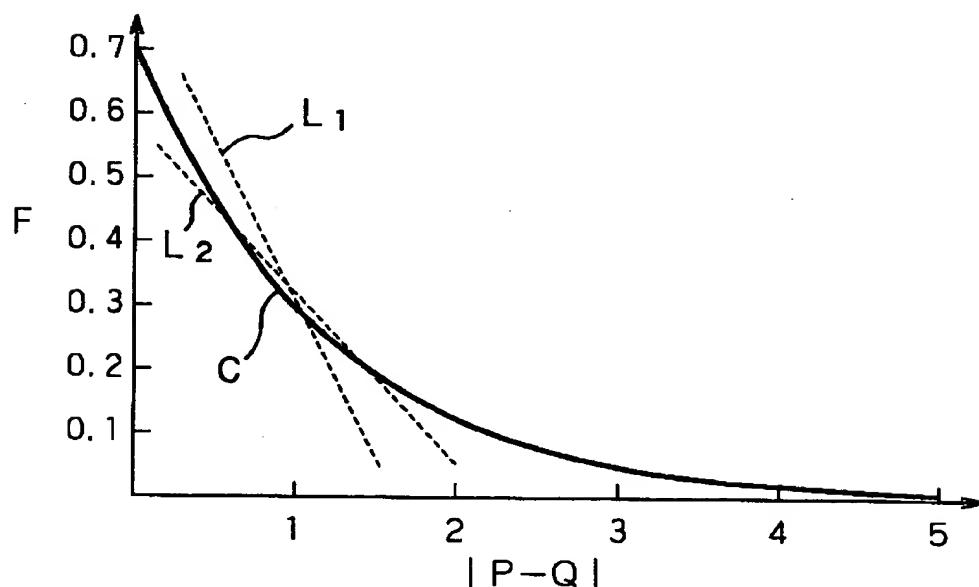
加算比較選択回路の構成ブロック図

【図10】



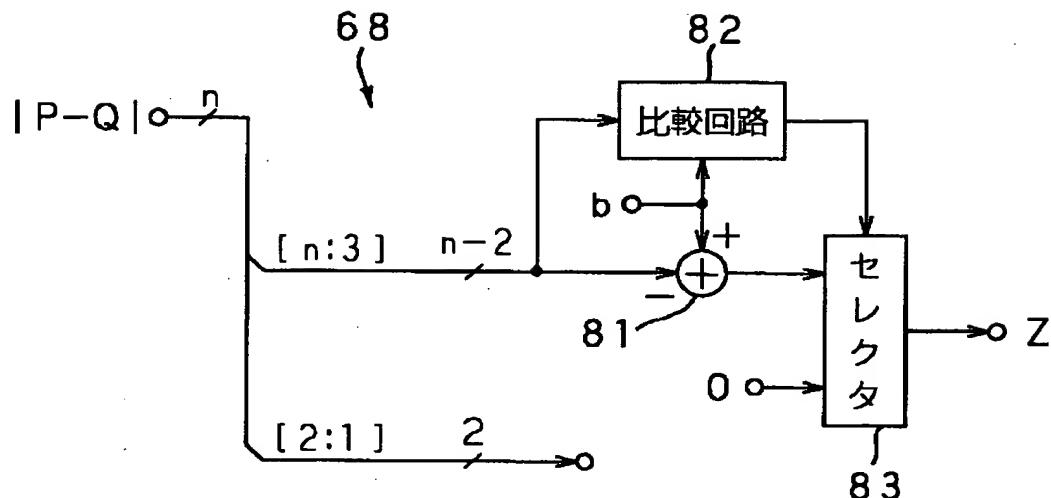
絶対値算出回路の構成ブロック図

【図11】



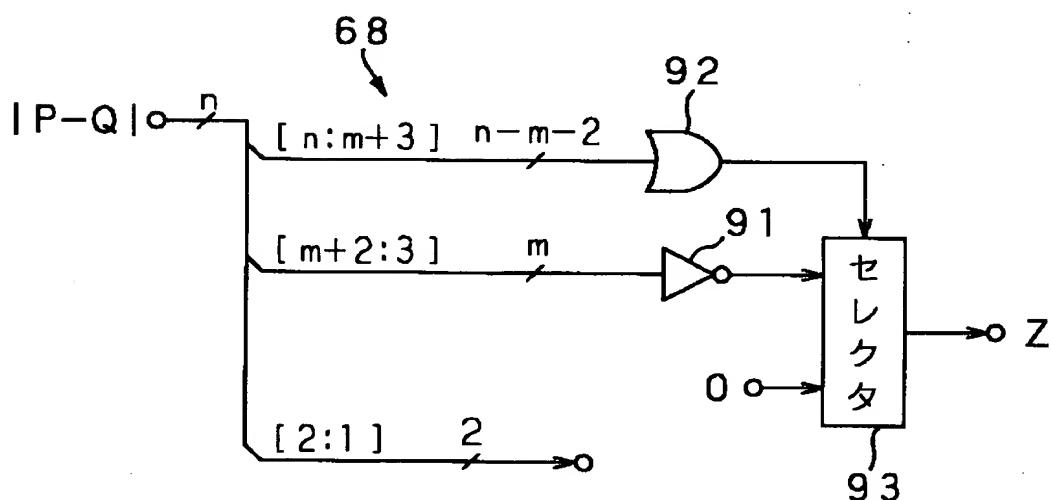
log-sum補正の説明図

【図12】



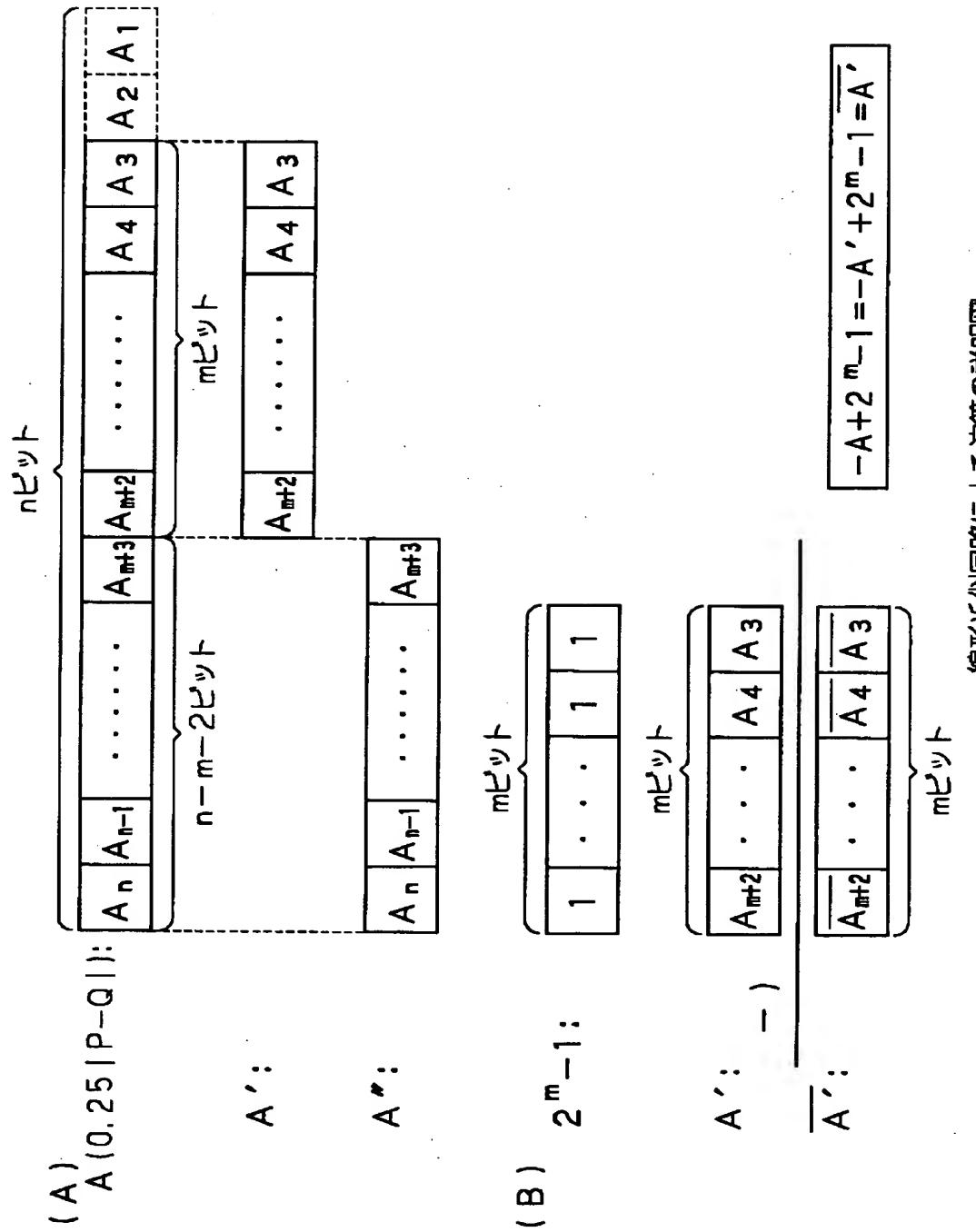
線形近似回路の構成ブロック図

【図13】



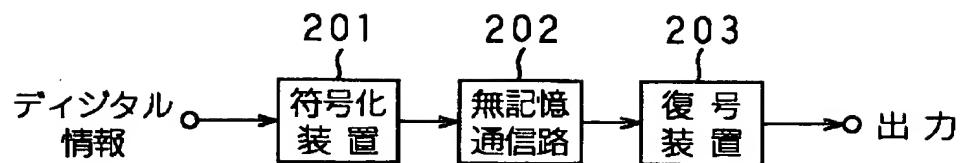
線形近似回路の構成ブロック図

【図14】



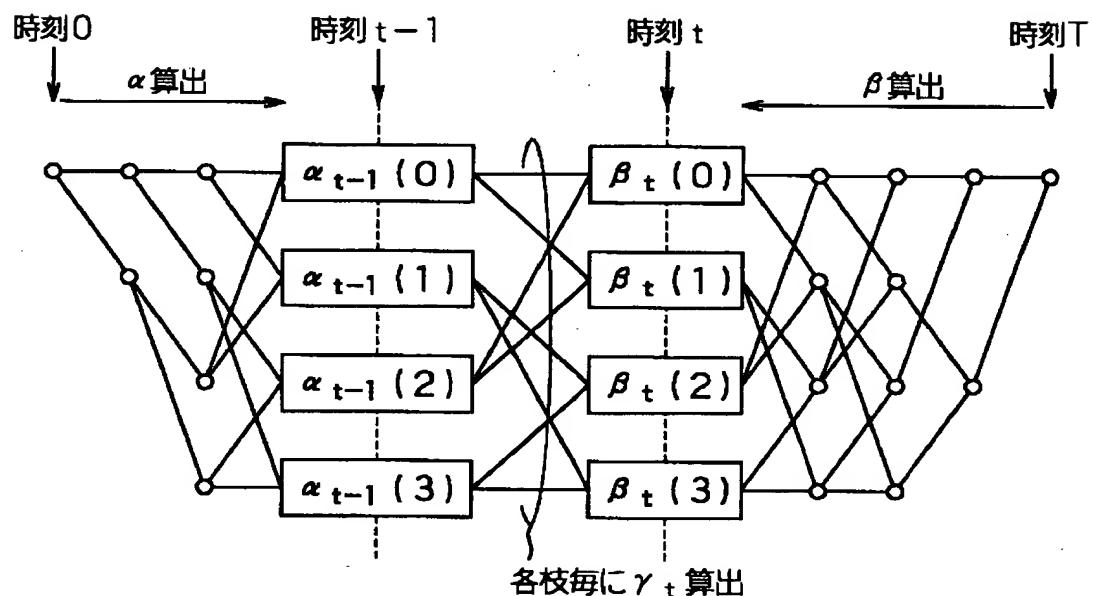
線形近似回路による演算の説明図

【図15】



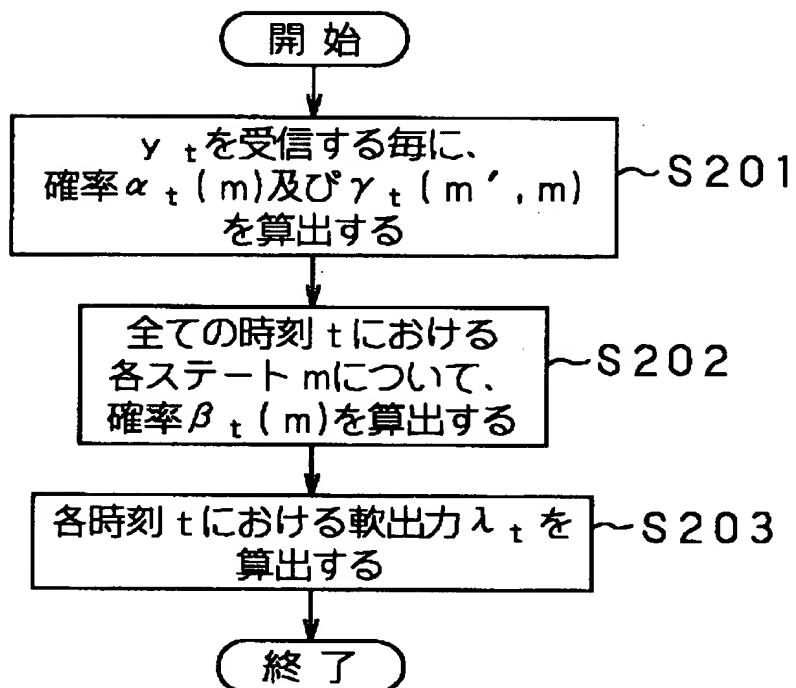
通信モデルの構成ブロック図

【図16】



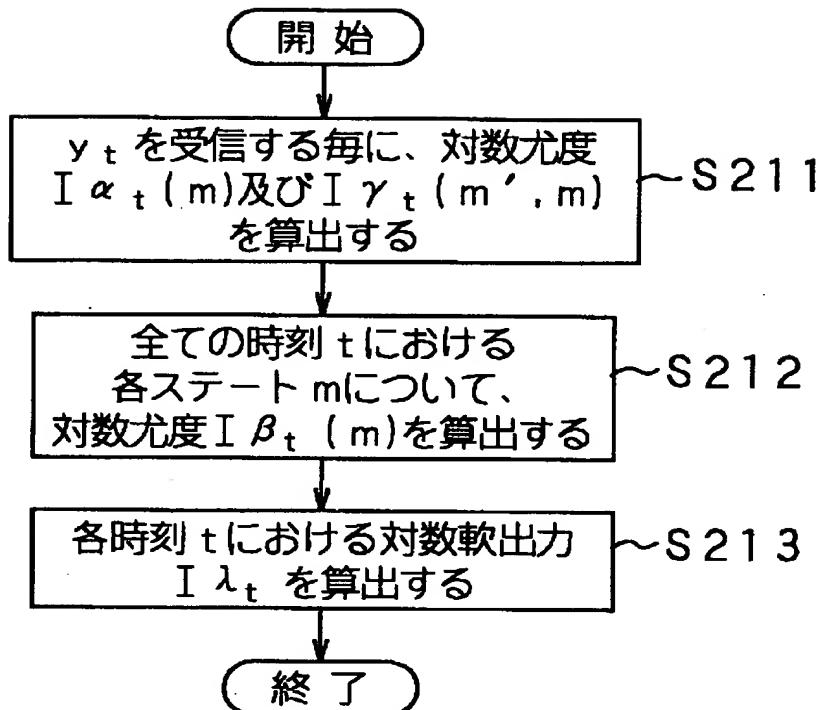
トレリスの説明図

【図17】



復号装置における一連の処理工程

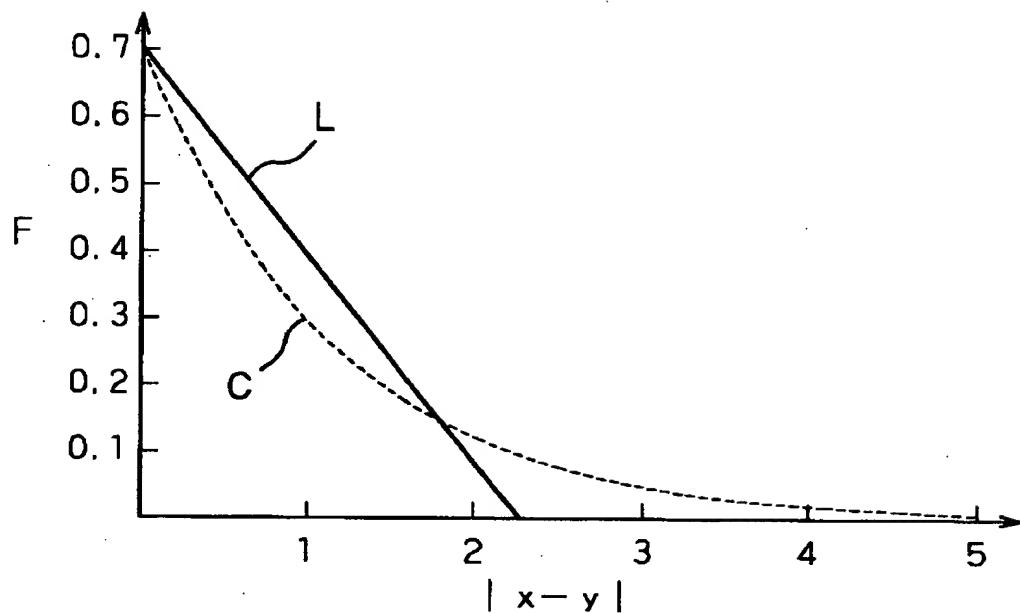
【図18】



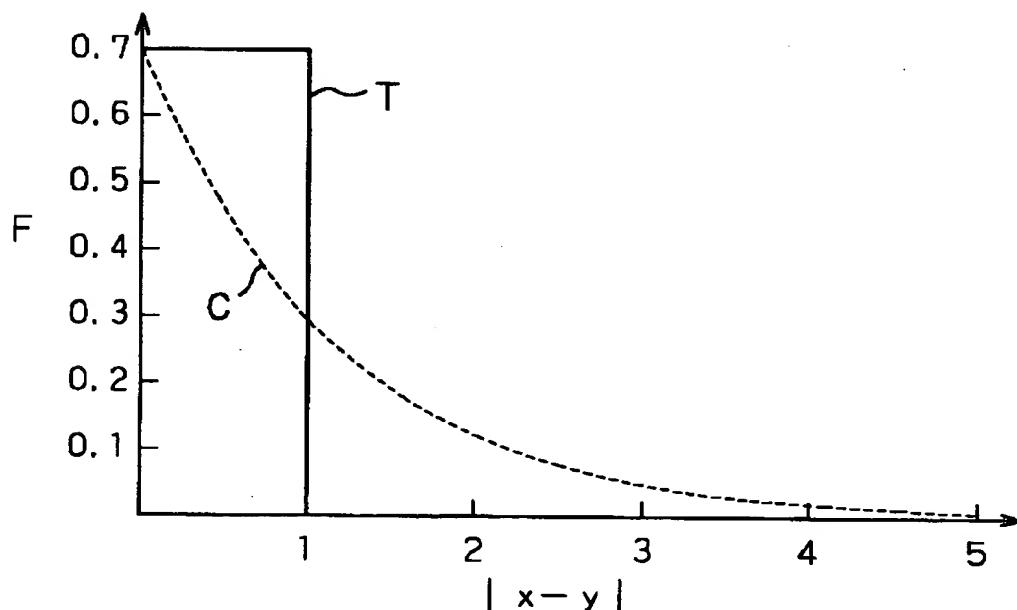
復号装置における一連の処理工程

【図19】

(A)



(B)



log-sum補正の説明図

【書類名】 要約書

【要約】

【課題】 速度を重視した線形近似により log-sum補正を行い、性能を劣化させることなく、高速化を図る。

【解決手段】 復号装置は、対数尤度を与えるために追加され、変数に対する1次元の関数で表される補正項を線形近似により算出する線形近似回路68を備える。線形近似回路68は、関数 $F = -a |P - Q| + b$ の傾きを表す係数-aと切片を表す係数bとを2のべき乗を用いて表現して線形近似法によるlog-sum補正を行い、補正項を算出する。この線形近似回路68は、係数aを -2^{-k} で表現し、係数bを $2^m - 1$ で表現したとき、絶対値データ $|P - Q|$ の下位1ビット目から下位kビット目までを切り捨ててビットシフトし、下位k+1ビット目から下位m+kビット目までのmビットをインバータ91により反転する。

【選択図】 図13

出願人履歴情報

識別番号 [000002185]

1. 変更年月日 1990年 8月30日

[変更理由] 新規登録

住 所 東京都品川区北品川6丁目7番35号

氏 名 ソニー株式会社